Calhoun: The NPS Institutional Archive
DSpace Repository

Theses and Dissertations 1. Thesis and Dissertation Collection, all items

1997-09

Modal testing and analysis of the NPS Space Truss.

Andberg, Brent K.

Monterey, California. Naval Postgraduate School

http://hdl.handle.net/10945/8683

Downloaded from NPS Archive: Calhoun

Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

DUDLEY KNOX LIBRARY

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

http://www.nps.edu/library

NPS ARCHIVE
1997
ANDBERG, B.

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

MODAL TESTING AND ANALYSIS OF THE NPS SPACE TRUSS

by

Brent K. Andberg

September, 1997

Thesis Advisor: Brij N. Agrawal
Co-Advisor: Gangbing Song

Thesis
A444215

Approved for public release; distribution is unlimited

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY CA 93943-5101

REPORT DOCUMENTATION PAGE	Form Approved OMB No. 0704-0188

lic reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, ching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send ments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to hington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA)2-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

GENCY USE ONLY (Leave blank)	2. REPORT DATE September 1997	3. REPORT TYPE AND DATES COVERED Master's Thesis

ITLE AND SUBTITLE DAL TESTING AND ANALYSIS OF THE NPS SPACE TRUSS	5. FUNDING NUMBERS
AUTHOR(S) berg, Brent K.	
ERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) al Postgraduate School nterey, CA 93943-5000	8. PERFORMING ORGANIZATION REPORT NUMBER
PONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING / MONITORING AGENCY REPORT NUMBER

SUPPLEMENTARY NOTES

views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of ense or the U.S. Government.

DISTRIBUTION / AVAILABILITY STATEMENT	12b. DISTRIBUTION CODE
roved for public release; distribution unlimited.	

ABSTRACT *(maximum 200 words)*

This thesis deals with modal testing and analysis of the Naval Postgraduate School (NPS) Space Truss. A e element model (FEM) was developed for the truss using a MATLAB™ program called NRLFEMI (developed at Naval Research Laboratory). Analytical predictions of the natural frequencies for this 3.76 meter by 0.35 meter ision structure were calculated using the NRLFEMI code. These calculated natural frequencies were then pared to experimental data collected during modal testing of the truss in the NPS Dynamics and Control oratory. Through analysis, the predicted results of the measurements (from the FEM) were satisfactorily correlated he experimentally obtained results, validating the FEM program. Additionally, a technology demonstration of r Bragg Grating Sensors (FBGSs) was performed. These laser etched, fiber optic sensors are ideally suited for -time evaluation of load, strain, vibration, and other health monitoring functions of structures.

SUBJECT TERMS al Testing, Modal Analysis, Fiber Bragg Grating Sensors, Finite Element Models			15. NUMBER OF PAGES 135
			16. PRICE CODE
SECURITY SSIFICATION OF ORT lassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFI- CATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

Approved for public release; distribution is unlimited

MODAL TESTING AND ANALYSIS OF THE NPS SPACE TRUSS

Brent K. Andberg
Captain, United States Marine Corps
B.S., United States Naval Academy, 1991

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ASTRONAUTICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 1997

NPS ARCHIVE
1997.09
ANDBERG, E.

Thesis
A44425
c.2

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY CA 93943-5101

ABSTRACT

This thesis deals with modal testing and analysis of the Naval Postgraduate School (NPS) Space Truss. A finite element model (FEM) was developed for the truss using a MATLAB™ program called NRLFEMI (developed at the Naval Research Laboratory). Analytical predictions of the natural frequencies for this 3.76 meter by 0.35 meter precision structure were calculated using the NRLFEMI code. These calculated natural frequencies were then compared to experimental data collected during modal testing of the truss in the NPS Dynamics and Control Laboratory. Through analysis, the predicted results of the measurements (from the FEM) were satisfactorily correlated to the experimentally obtained results, validating the FEM program. Additionally, a technology demonstration of Fiber Bragg Grating Sensors (FBGSs) was performed. These laser etched, fiber optic sensors are ideally suited for real-time evaluation of load, strain, vibration, and other health monitoring functions of structures.

TABLE OF CONTENTS

LIST OF FIGURES

I. INTRODUCTION

Spacecraft launch cost is the major part of the total cost of any space mission. This often prohibitive cost of launching payloads into orbit has driven engineers and scientists to develop lighter construction materials and weight saving designs, without compromising the dynamic stability and strength required for mission accomplishment. With these new materials and designs, however, comes the added challenges of modeling, measuring, and controlling the vibration of advanced, lightweight, space structures.

The Naval Postgraduate School (NPS) Space Truss provides a key platform for the development and test of models and modeling techniques. Once a Finite Element Model (FEM) has been developed for the NPS Space Truss, it can be experimentally verified in the laboratory. Specifically, using accelerometers attached to the truss's nodes and exciting the truss via an impulse hammer (with a built-in force transducer) striking a node, the frequency response function can be obtained. The resulting Frequency Response Functions (FRFs) can be compared to the FEM using a computer tool such as X-MODAL thereby validating the FEM. Our goal should be to develop an accurate FEM (which could subsequently be used to predict other truss characteristics that may be unfeasible to produce in the laboratory), and then to validate that FEM through modal testing and analysis.

Once a model has been verified as accurate, control methods can be applied and tested as well. A new and advanced dynamic and strain detection system is comprised of Fiber Bragg Grating Sensors (FBGS). Employing light weight, low power fiber optics, this recently developed technology provides the spacecraft designer with a simple, yet highly accurate measurement tool for detecting both static and dynamic strain on the structure. Once detected via FBGS, a control system may be implemented to bring the structure's strain condition to within acceptable levels. Current applications include long-term static strain sensing, dynamic strain sensing (frequencies are presently low, about 50 Hz, but faster systems will soon be available), temperature and pressure sensing, magnetic and electric field sensing, and chemical sensing.

Ultimately, the NPS Space Truss provides an opportunity for comparing modeled modal analysis with actual experimental results. Employing new technology in the form of FBGSs, feedback from the fiber sensors can be provided to actuators for active control.

II. ANALYTICAL MODEL OF THE NPS SPACE TRUSS

A. DETAILED TRUSS DESCRIPTION

1. Elements

The NPS Space Truss structure is composed of twelve cubic bays assembled from a combination of 161 elements which begin and terminate in an Aluminum node ball. There are a total of 52 node balls (see Figure 2) comprising the truss. The structure is approximately 3.76 meters in length, 0.35 meters wide, and 0.7 meters tall (from the base plate).

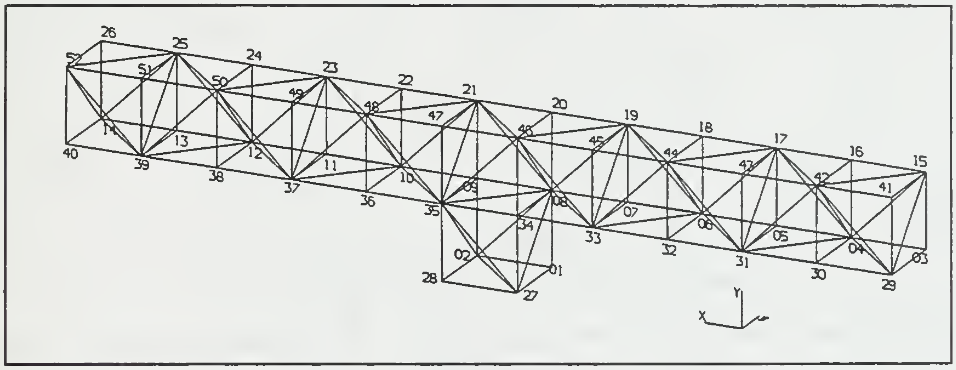


Figure 1. NPS Space Truss (with numbered nodes)

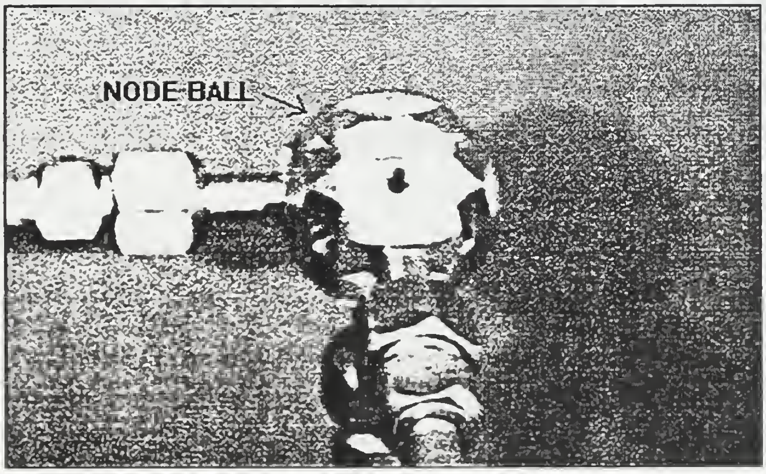


Figure 2. Node Ball

These twelve cubic bays are a combination of battens/longerons and diagonals. Longerons run down the length of the structure, battens compose the vertical elements, and diagonals run diagonally from one line of longerons to an adjacent line. Collectively, all of the afore mentioned elements will be referred to as struts. Each strut is made of homogeneous Aluminum, and is composed of several parts: the tube, outer sleeve, bolt, standoff, and nut (see Figure 3). Additionally, the tube is fastened to the outer sleeve with epoxy and then a pin is driven through the sleeve and tube. Each strut begins and terminates in an Aluminum node ball.

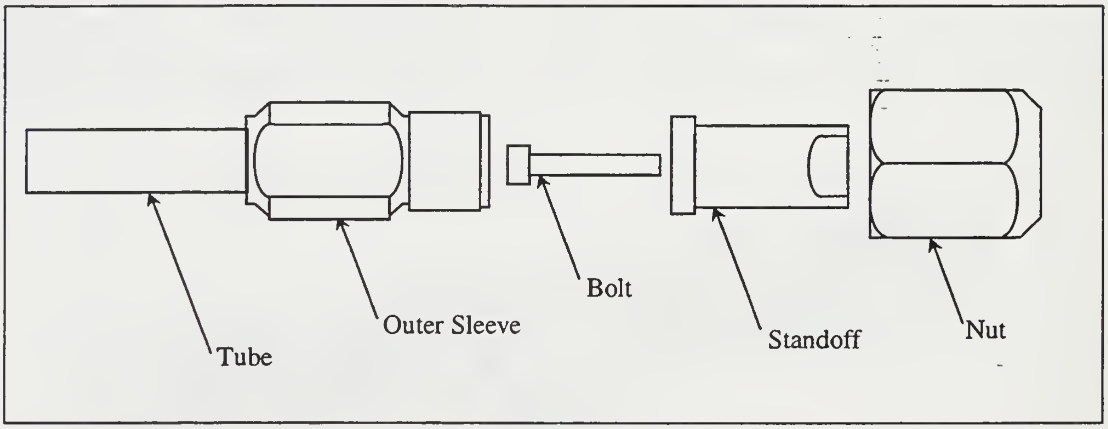


Figure 3. Schematic of Strut Terminating End

Note: 1) pin and epoxy (not shown) connect outer sleeve with the strut tube; 2) not drawn to scale.

The struts can be modeled as rod elements. Rods (or bars) can be defined as elements whose geometry is such that the longest dimension of the bar is straight and the greatest dimension of the cross section is small compared to the length. A rod is an axial member with an internal axial force only, known as a two-force member. [Ref. 3] Figure 4 is a basic schematic for a rod element.

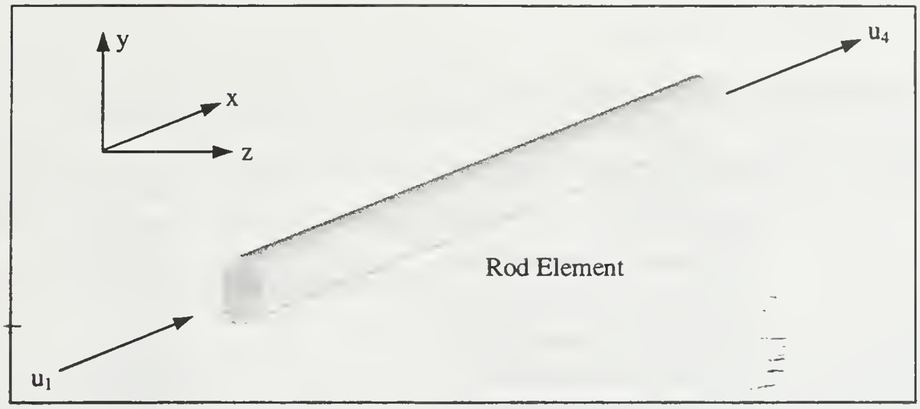


Figure 4. Rod Element

The elemental stiffness and mass matrices for the rod model, respectively, are shown in Equations (2.1) and (2.2).

$$k_{rod} = \begin{array}{c} \begin{array}{cccccc} u_1 & u_2 & u_3 & u_4 & u_5 & u_6 \end{array} \\ \begin{bmatrix} \frac{EA}{L} & 0 & 0 & \frac{-EA}{L} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{-EA}{L} & 0 & 0 & \frac{EA}{L} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array} \qquad (2.1)$$

$$m_{rod} = \frac{\rho AL}{12} \begin{bmatrix} 5 & 0 & 0 & 1 & 0 & 0 \\ 0 & 6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 6 & 0 & 0 & 0 \\ 1 & 0 & 0 & 5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 6 \end{bmatrix} \qquad (2.2)$$

In equation (2.1) and (2.2):

A ≡ cross-sectional area

E ≡ elastic modulus

L ≡ length of element

ρ ≡ density

2. Truss Construction

The mass breakdown of the individual parts of the truss is as follows:

Quantity:	**Part name:**	**Individid. Mass:** (kg)
52	node balls	0.06625
100	longerons (unassembled)	0.01385
100	longerons (assembled)	0.04475
61	diagonals (unassembled)	0.02125
61	diagonals (assembled)	0.05215
322	bar end assemblies (each)	0.01545
322	screws (minus heat shrink & vibe tight)	0.60697 (total weight)
Assembled Truss	(bare, sum of above, assembled parts)	**11.7081**
Assembled Truss	(bare, actual measured mass)	**11.750**
Base plate	(not included in calculated or meas. mass)	7.30

Table 1. Truss Mass Breakdown

The NPS Space Truss is precisely assembled in the following manner. After each part was fabricated, the individual struts were assigned identifying serial numbers. These serial numbers were printed on tabs and attached to their respective members and covered with a transparent piece of heat shrink. In addition, each end assembly has the suffix of its strut's serial number etched on it in order that each two end assemblies remain permanently paired with their respective strut. End assemblies, without their struts, were first attached to their respective node balls. The node balls are aluminum spheres, approximately 38.7 mm in diameter (see Figure 2). Each node ball has eighteen connection points for attaching struts with end assemblies and for attaching thumb screws. A torque, socket wrench, set to 44 in-lbs., and fitted with a 9/64th inch hex head was used to tighten the #8-32 screws which fasten the end assembly to the node ball. There are 322 of these screws in the whole truss assembly (basically, two for each strut). Each screw is prepared with heat shrink/vibe tight, which restricts a screw's ability to loosen itself during prolonged, high frequency vibrations. After attaching the end assemblies to the node balls, the end assemblies were paired with their struts. An 11/16th torque wrench, set to 70 in-lbs. (supplied by NRL, and stored in proximity to the truss) was used in conjunction with an open, ½ inch crescent wrench to tighten down the end assemblies on the struts. To aid the engineer in threading and tightening the end assemblies, *Castrol Braycote* 601EF (an aerospace, flight qualified lubricant) was applied to the strut's threading. Recall from Table 1, the Truss Mass Breakdown, that the bare,

measured mass of the assembled truss was 11.750 kg while the truss's bare, calculated assembled mass was only 11.708 kg. This 0.26% increase in mass is probably due to the added masses of the heat shrink (on all 161 struts), the vibe tight (on all 322 screws), and the *Castrol Braycote*. Regardless, this extra mass is a negligible amount and for the purposes of modeling, the bare, calculated, assembled truss mass will be used.

B. DYNAMIC STIFFNESS TESTING

1. Introduction

In the case of the truss struts, we are interested in the effective axial stiffness from node-point to node-point (the center of a node ball is effectively a node-point). In other words, the effective axial stiffness of a strut is from the center of one node ball, to the next node ball, along the length of a truss element. The stiffness of individual parts may be reasonably calculated however, their combined, effective stiffness is not as easily obtained. A dynamic measurement procedure was devised by Robert Craig Waner at the Naval Research Lab for just such a measurement (August, 1995). This dynamic procedure is similar to that of Hallauer and Lamberson [Ref. 2] and will be discussed later. For now, the effective axial stiffness of a rod element may be defined as follows:

$$k_{eff} = \left(\frac{AE}{L}\right)_{eff} \tag{2.3}$$

where: A ≡ cross-sectional area

E ≡ Young's modulus

L ≡ length

2. Analytical Development

Struts can effectively be modeled as a springs with specific stiffness values (k_{eff}). The dynamic test for effective axial stiffness will incorporate a system of two point masses (m_1 and m_2) connected by a linear spring (k_{eff}) as illustrated in Figure 5 (x_1 and x_2 are a global coordinate system).

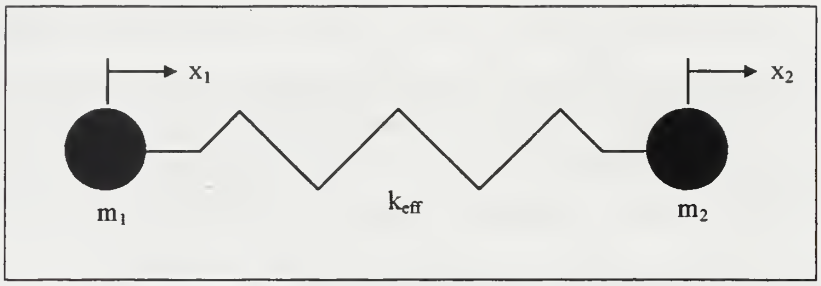


Figure 5. Schematic of Free-Free System

When we apply Newton's second law to the system we will arrive at the following equations of motion:

$$m_1\ddot{x}_1 + k_{eff}x_1 - k_{eff}x_2 = 0 \tag{2.4a}$$

$$m_2\ddot{x}_2 + k_{eff}x_2 - k_{eff}x_1 = 0 \tag{2.4b}$$

which results in the following matrix:

$$\begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix}\begin{Bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \end{Bmatrix} + \begin{bmatrix} k_{eff} & -k_{eff} \\ -k_{eff} & k_{eff} \end{bmatrix}\begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \tag{2.5}$$

Assume a harmonic solution of the form in the following equation:

$$\{x\} = \{x_0\}\cos(\omega t + \phi) \tag{2.6}$$

where: $\{x_0\} \equiv$ 2 by 1 vector of time-independent amplitudes

$\omega \equiv$ undamped natural frequency of system

$\phi \equiv$ phase angle

If we now substitute Equation **(2.6)** and its derivatives into Equation **(2.5)**, we will arrive at the new matrix below:

$$\begin{bmatrix} (-m_1\omega^2 + k_{eff}) & -k_{eff} \\ -k_{eff} & (-m_2\omega^2 + k_{eff}) \end{bmatrix}\begin{Bmatrix} x_{0_1} \\ x_{0_2} \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \tag{2.7}$$

In this new matrix, equation **(2.7)**, the vector $\{x_0\}$ is the nullspace of the left-hand matrix. Since every matrix has a null space, ω must be chosen such that the left-hand matrix has a nullspace. This dictates that the left-hand matrix must be singular, and therefore, its determinant must be equal to zero.

When we take the determinant of the left-hand matrix in equation **(2.7)**, and set it equal to zero, we are left with the following expression:

$$m_1 m_2 \omega^4 - k_{eff} m_1 \omega^2 - k_{eff} m_2 \omega^2 = 0 \qquad \textbf{(2.8)}$$

Solving equation (**2.8**) for ω^2:

$$\omega^2 = 0 \text{ (rigid body mode)} \qquad \textbf{(2.9a)}$$

$$\omega^2 = \frac{k_{eff}(m_1 + m_2)}{m_1 m_2} \qquad \textbf{(2.9b)}$$

Extracting k_{eff} from equation (**2.9b**) gives us the following expression for the effective stiffness:

$$k_{eff} = \frac{m_1 m_2 \omega^2}{(m_1 + m_2)} \qquad \textbf{(2.10)}$$

3. Stiffness Experimental Implementation

To determine the effective axial stiffness of individual struts, an experiment was set up which included a strut (with terminating assemblies) and a single node ball bolted between two weights suspended by turnbuckles and two wires anchored at points ten feet above the floor (see Figure 6). An Endevco Model 61-500 accelerometer was attached to one of the weights (S/N AE50, sensitivity 481 mV/g). The other weight was excited with a single impulse from a Kistler Type 9722A500 Impulse Hammer (S/N C46195). The outputs from both the accelerometer and the impulse hammer were fed into a Piezotronics Model 483A07 ICP Signal Conditioner which was then fed into a Hewlett Packard, HP35670A, Dynamic Signal Analyzer (S/N 3431A01574). Finally, output from the HP Dynamic Signal Analyzer was saved to diskette and analyzed on a PC using MATLAB.

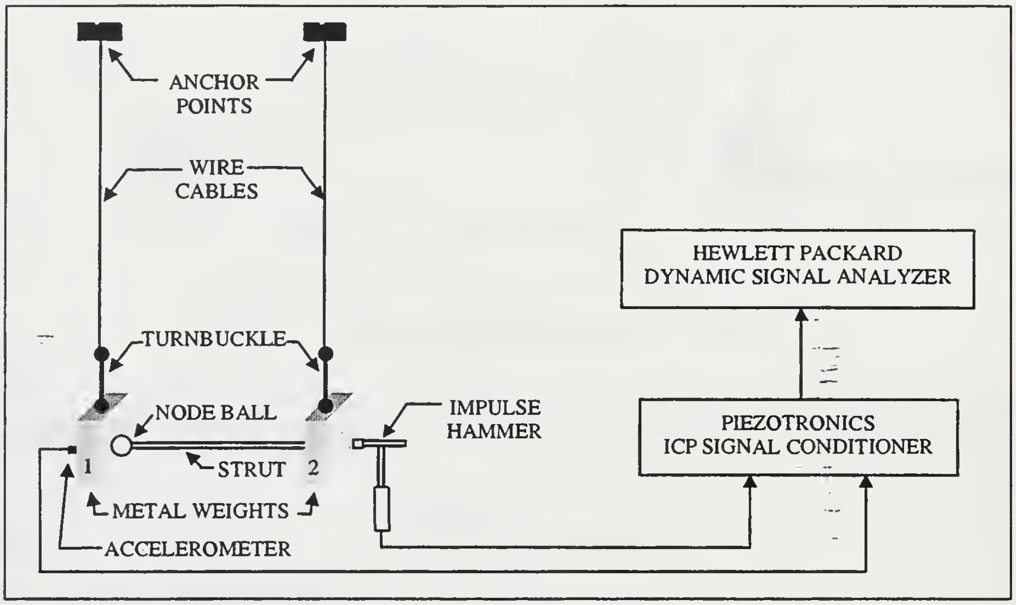


Figure 6. Effective Axial Stiffness Test Setup

The procedure consists of tapping the metal block (weight) on the side opposite the accelerometer with the impulse hammer. The velocity vector of this tap should be parallel to the longitudinal axis of the strut. This impulse will excite the strut to its natural frequency. The accelerometer attached to the opposite metal block will sense the vibration. Using the Hewlett Packard Dynamic Signal Analyzer attached to the Piezotronics ICP Signal Conditioner, a frequency response function (FRF) was generated. Referencing the largest peak in this FRF, a corresponding natural frequency can be determined. Recall that in our analytical development, equation **(2.9a)** predicted a rigid body mode ($\omega^2 = 0$). In the experimental model, we only approximate a free-free system (we have the wire cables to contend with, however negligible they may be) where as the analytical model is a true, free-free system. Once we know ω, the natural frequency, we need only know the total mass of the metal end blocks, m_1 and m_2, to determine the effective stiffness, k_{eff}. m_1 consists of one metal block, the accelerometer, the node ball, one half of the tube mass, and an end assembly (outer sleeve, bolt, standoff, and nut; see Figure 3). m_2 consists of one metal block, one half of the tube mass, and an end assembly. We now have the values necessary to calculate the effective stiffness (see equation **(2.10)**).

4. Stiffness Experimental Results

Using the dynamic stiffness test, five different battens/longerons and five different diagonals were tested. Each element was tested five times to develop an average for that specific element. Then the five longeron averages and the five diagonal element averages were averaged to develop an effective axial stiffness for that type of element. The following two tables display the results, complete with the average effective stiffness. Additionally, a static pull test was preformed on the same struts. Both values are present for comparison. The values for the dynamic stiffness test will be the values used in the Finite Element Model, however. In the following tables (Tables 2 and 3), effective stiffness determined by the dynamic test is $\mathbf{k}_{\mathbf{eff}}$, and effective stiffness determined by the static pull test is $\mathbf{k}_{\mathbf{sta}}$ (both tests were conducted at the Naval Research Lab). Note that the standard deviation for the longeron stiffness test is only a half a percent and the standard deviation for the diagonal elements is on 1.08%.

Battens/Longerons						
Number	Serial #	f (Hz)	ω (rad/sec)	keff (N/m)	keff (lb/in)	ksta (lb/in)
1	1-C-003	374.0	2349.911	5.19E+06	29609	29589
2	11-E-185	373.0	2343.628	5.16E+06	29451	29227
3	11-K-191	372.0	2337.345	5.13E+06	29293	32872
4	11-D-184	374.0	2349.911	5.19E+06	29609	30451
5	11-F-186	373.0	2343.628	5.16E+06	29451	28956
			average =	5.16E+06	29482	30219
			std. dev. =	2.31E+04	132	1587
			std. dev./ave. =	0.45%	0.45%	5.25%

Table 2. Batten/Longeron Effective Stiffness

Diagonal Elements						
Number	Serial #	f (Hz)	ω (rad/sec)	keff (N/m)	keff (lb/in)	ksta (lb/in)
1	10-R-177	301.5	1894.380	3.38E+06	19280	17852
2	6-N-089	303.0	1903.805	3.41E+06	19472	18866
3	10-S-178	304.0	1910.088	3.43E+06	19601	18334
4	10-T-179	303.5	1906.947	3.42E+06	19537	19277
5	10-P-175	300.0	1884.956	3.34E+06	19089	18041
			average =	3.40E+06	19396	18474
			std. dev. =	3.67E+04	210	590
			std. dev./ave. =	1.08%	1.08%	3.19%

Table 3. Diagonal Effective Stiffness

C. BUILDING THE ANALYTICAL MODEL

1. Description Of NRLFEMI

NRLFEMI is a FEM written in MATLAB code specifically designed for the NPS Space Truss. MATLAB interprets this code, developed at the NRL, utilizing script files which define specific properties about the truss. Some of the characteristics of NRLFEMI include:

- Menu-driven through a graphical user interface.
- Make geometric and material property data entry simple.
- Other capabilities:
  - Compute Eigenstructure (Eigenvalues and Eigenvectors)
  - Static graphical display of structure
  - Compute and plot frequency response functions
  - Save data to files for later recall
  - Be able to store all stiffness submatrices compactly for the purposes of model updating
  - Animate normal mode shapes in MATLAB

Some of the assumptions made by NRLFEMI are that the truss bays are to be cubic. Deviation from this geometric configuration will cause a fatal error when creating the mass and stiffness matrices. Also, the elemental mass matrices in the code are constructed using a local coupled mass matrix, as opposed to local lumped or consistent mass matrices. Equations **(2.1)** and **(2.2)** describe the elemental stiffness and mass matrices for the rod model, respectively. The NRLFEMI program is presently available on several computers in the Dynamics Lab at the Naval Postgraduate School. The code used in the program and a better description of how to modify the truss properties are included in the appendices. For now, a brief description of the program's implementation and its results are all that is necessary.

2. FEM Implementation and Results

We have already collected the truss properties required to build an accurate model of the structure. Specifically, we know the masses of different elements, we know the effective axial stiffness values for the longerons and diagonal elements, and we know how the truss is constructed (which element connects to which node, etc.). After we input

this data into the NRLFEMI program, we get the natural frequencies of the NPS Space Truss. Although these frequencies provide a good estimate to compare actual frequencies with, there are several limitations to the model.

The first such limitation comes from the actual truss's mass deviations from the calculated truss mass. Referring back to Table 2.1, Truss Mass Breakdown, we know that there was some additional, unaccounted for mass on the actual weighed truss. This was attributed to vibe tight, heat shrink, and lubrication. It should now be pointed out that the bare truss is not the same as the actual, tested truss. Likewise, the bare truss's mass is not the mass used in determining the truss characteristics for NRLFEMI. Specifically, the test mass of the NPS Space Truss includes the combined mass of eight accelerometers and 40 aluminum dummy masses (see Figure 7).

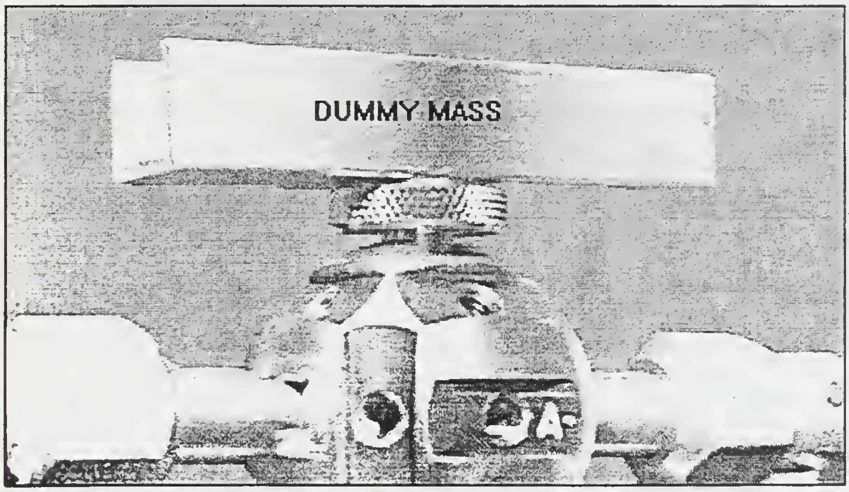


Figure 7. Dummy Mass on Node Ball

The dummy mass's and accelerometer's mass are both 11.2 grams. Although these masses are precisely accounted for, it is in their application that the error resides. Both dummy masses and accelerometers (total of 48) are attached to the truss with Petro Wax (an adhesive discussed in a latter chapter). The combined effect of all these masses: heat shrink, vibe tight, lubrication, and adhesives, is enough to change the characteristics of the actual, experimental truss, from the modeled one. Unfortunately, there is no accurate and feasible way to measure these extra masses independently.

A second limitation manifests itself in damping of the truss. Each of the eight accelerometers has a connection cable (for power and the returned signal) that connects it to the signal conditioner. These cables have a negligible mass and don't effect the mass

distribution of the truss, however thy may provide from 1% to 2% damping. The combined effects of extra mass on the experimental truss and the unwanted damping will result in a noticeable difference between the NRLFEMI natural frequencies, and the actual, experimental frequencies.

The following table, Table 4, NPS Space Truss Natural Frequencies, corresponds to the first 20 natural frequencies, as computed by NRLFEMI. Table 5, NRL Truss Natural Frequencies, is also displayed for comparison. The original NRL truss had steel node balls in place of the Aluminum node balls present on the NPS Space Truss. The reader will notice that the extra mass at the nodes forced the natural frequencies of the NRL truss lower then those of its NPS counterpart.

Number	ω_n(rad/s)	**frequency**(Hz)
1.00	92.01	14.64
2.00	102.14	16.26
3.00	191.06	30.41
4.00	213.44	33.97
5.00	395.40	62.93
6.00	468.36	74.54
7.00	506.79	80.66
8.00	634.66	101.01
9.00	793.12	126.23
10.00	854.35	135.97
11.00	885.68	140.96
12.00	1246.87	198.44
13.00	1305.21	207.73
14.00	1442.56	229.59
15.00	1461.82	232.66
16.00	1616.49	257.27
17.00	1762.29	280.48
18.00	1788.50	284.65
19.00	1970.66	313.64
20.00	2206.76	351.22

Table 4. NPS Space Truss Natural Frequencies (calculated)

Number	ω_n(rad/s)	frequency(Hz)
1.00	79.24	12.61
2.00	88.65	14.11
3.00	166.23	26.46
4.00	185.71	29.56
5.00	343.14	54.61
6.00	404.34	64.35
7.00	435.82	69.36
8.00	549.16	87.40
9.00	689.26	109.70
10.00	742.51	118.17
11.00	769.08	122.40
12.00	1079.98	171.88
13.00	1139.06	181.29
14.00	1246.46	198.38
15.00	1255.79	199.87
16.00	1404.31	223.50
17.00	1527.16	243.06
18.00	1548.73	246.49
19.00	1704.14	271.22
20.00	1902.63	302.81

Table 5. NRL Space Truss Natural Frequencies (calculated)

III. EXPERIMENT AND ANALYSIS OF MODAL TESTING ON THE NPS SPACE TRUSS

A. EXPERIMENTAL SETUP

1. Development

The basic relationship when performing modal testing is:

$$\boxed{\text{RESPONSE}} = \boxed{\text{PROPERTIES}} \times \boxed{\text{INPUT}} \quad \text{[Ref. 7]}$$

In the case of the NPS Space Truss, we will be able to accurately determine the response and the input. From these two measurements and this relationship, we should be able to determine the truss properties (in our case, the natural frequencies), and then compare them with the calculated properties.

The equipment necessary for us to measure a response will be several accelerometers (in our case, eight), signal conditioners, a PC based analog to digital (AD) data acquisition system (dSpace™), and a host computer. In order to measure our input, we will need an excitation source, specifically, and impulse force hammer. In the setup used, a separate signal conditioner (one channel) was used with the impulse force hammer. This one channel, signal conditioner was also connected to the dSpace system (see Figure 8).

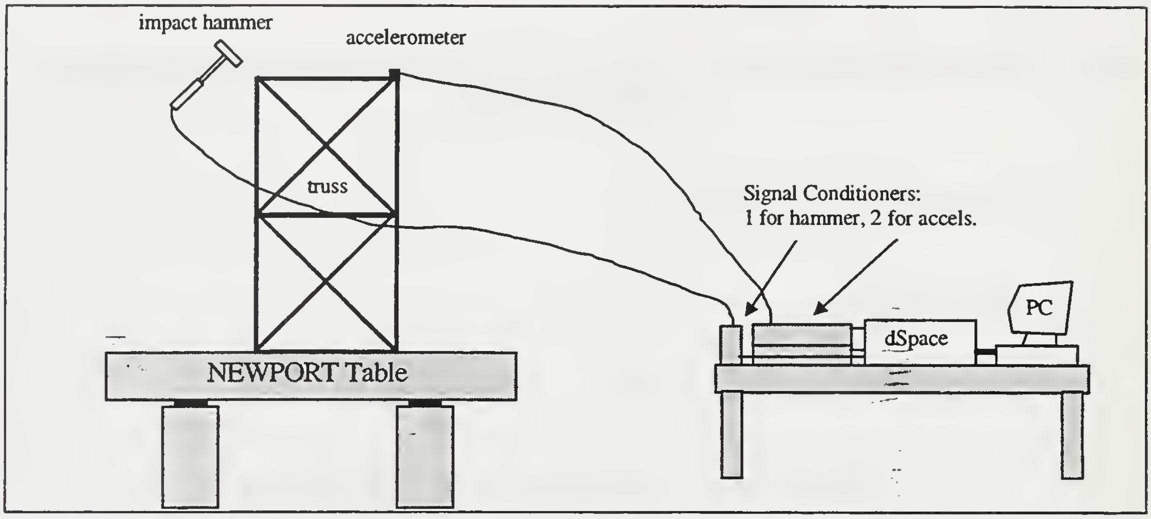


Figure 8. Experimental Setup

The NPS Space Truss is mounted on a Newport Vibration Control System Table. The Newport's 1000 lbs. table top is floated on a cushion of N_2 during testing. Each of its four pedestal legs contain a piston which, when charged with compressed air (or N_2 via a storage tank in our case), floats the table. The advantage of using such a mounting platform is that it filters out unwanted disturbance vibrations. Specifically, disturbances in the frequency range above 12 Hz are attenuated by more then 99% by the table. [Ref. 9] Accelerometers are highly sensitive measuring devices (sensitivities range from 100 to 500 mV/g). Any outside disturbance could corrupt the data being taken.

Since we only have eight accelerometers, we will need to run the experiment several times, while moving the accelerometers to different nodes between each test, in order to test the frequency response at every node. Ultimately, we will run the experiment six times, each time measuring the response on eight different nodes, to get a global picture of the truss's response. After successfully testing each node, we can alter the pattern of the accelerometer placement, tailoring the experiment to the type of data we wish to collect.

The final piece to our modal testing relationship is the input. Our input will be provided via an impulse force hammer. By striking a predetermined node such that the force is distributed equally along all three axis, we can excite the truss through its range of natural frequencies. Two nodes were selected as impulse hammer targets: node 41 and node 24 (see Figure 9). Nodes 41 and 24 were chosen so that we may excite the lower natural frequencies with a certain degree of force as well as the higher natural

frequencies, respectively. Node 41 is located on the extreme end of the truss where the first mode shape, corresponding to its first natural frequency, will have its greatest amplitude (nodes 15, 52, and 26 would work equally as well). Node 24 is located midway down the length of the truss, where the higher mode shapes have their greatest energy, and hence the higher natural frequencies.

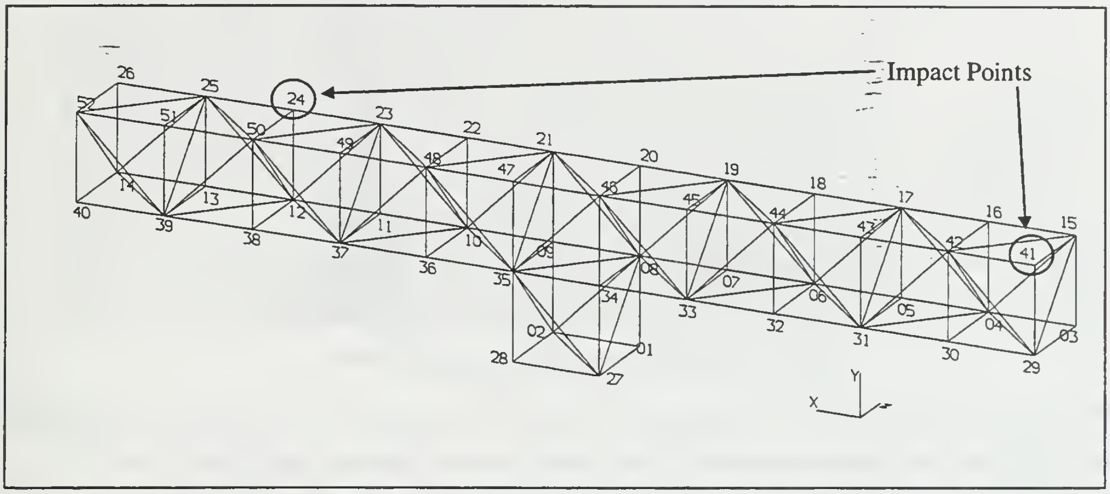


Figure 9. Impact Node Locations

2. Impulse Hammer Calibration

The eight accelerometers used in the modal testing of the NPS Space Truss were purchased new. As such, they arrived with current calibration certificates, stating that they their calibration would remain within an acceptable tolerance level for at least one year from the date of delivery. [Ref. 8] On the other hand, the PCB® Piezotronics Impulse Force Hammer used in the testing was last calibrated August 30, 1989. Therefore, a simple re-calibration method was devised and implemented in order to accurately run the modal testing.

The impulse hammer calibration test requires a suspended precision test mass (in this case, an 755.6 kg block of aluminum with polished, parallel faces), a calibrated accelerometer (only uniaxial is necessary), the impulse hammer to be tested, a signal conditioner, AD data acquisition system (dSpace), and a host computer with analysis software (see Figure 10).

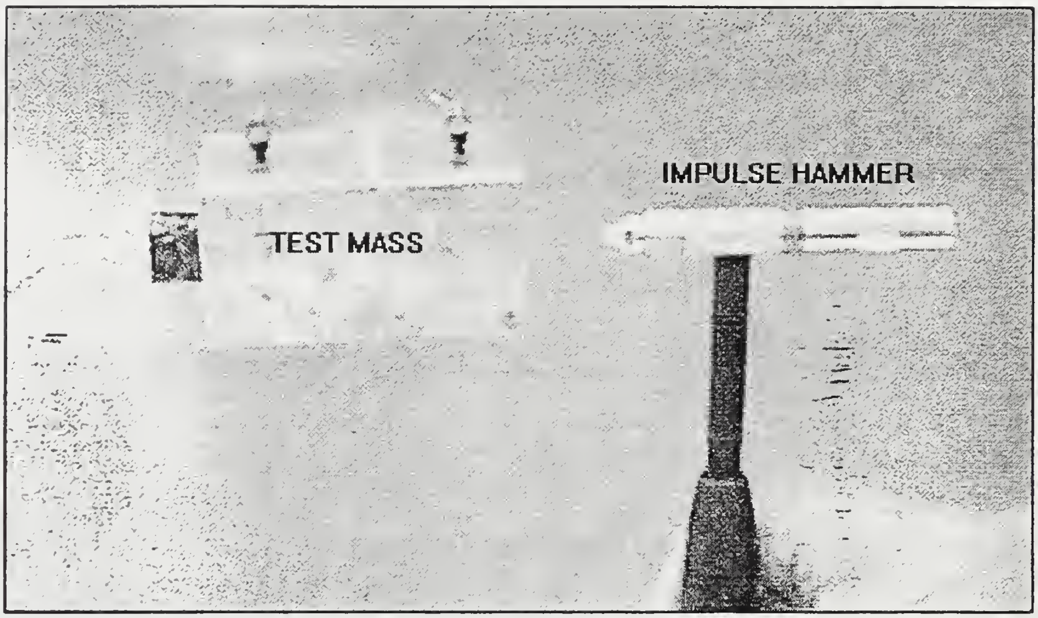


Figure 10. Impulse Hammer Calibration

Table 6 is an exact list of the equipment used in the test (the test mass was suspended using ordinary, monofilament fishing line, therefore its mass was negligible).

Item	Model & Serial #	Calibration value	Mass
Kistler accelerometer	Model # 8690C50 S/N C112865	10.132 g/V where g=9.807 m/s^2	11.2 g
PCB® Piezotronics Impulse Force Hammer	Model # 086B01 S/N 4144	(see test results)	n/a
Aluminum test mass	n/a	n/a	755.6 g (bare), 767.4 g (w/ accel. and adhesive)
Accel. signal condit.	Model # 5124A S/N C74930	n/a	n/a
PCB® Piezotronics Signal Conditioner	Model # 484B S/N 2086	n/a	n/a

Table 6. Impulse Hammer Test Materials

The test uses the basic Newtonian equation for force:

$$F = ma \tag{3.1}$$

The force is imparted by the impulse hammer onto the mass and both force and acceleration are measured by the impulse hammer and the accelerometer, respectively. Again, as in the dynamic stiffness testing, we approximate a free-free system. In this system, we will strike the mass, perpendicular to the force of gravity. Since the mass is suspended from a height (approximately 15 feet) by relatively massless wires (fishing

line), there is no resistance, or friction, along the direction of the impact vector for short distances. Ultimately, we have one equation, equation **(3.1),** and one unknown, *F*. To effectively employ Newton's equation, we will add the necessary conversion factors to come up with equation **(3.2)**.

$$k_f \cdot V_f = m \cdot k_a \cdot V_a \qquad \textbf{(3.2a)}$$

Rearranging the terms in equation **(3.2a)** gives us equation **(3.2b)** which yields the hammer's calibration value or sensitivity.

$$k_f = \frac{m \cdot k_a \cdot V_a}{V_f} \qquad \textbf{(3.2b)}$$

where: k_f ≡ calibrated hammer sensitivity in N/V
m ≡ test mass (=767.4 grams)
k_a ≡ accelerometer calibration (=10.132 g/V)
V_a ≡ accelerometer voltage reading
V_f ≡ impulse hammer voltage reading

Using this relationship, several tests were conducted and an average was generated for the impulse hammer sensitivity. Prior to each test, any swinging of the mass was damped by hand, and the mass itself was leveled using an ordinary carpenter's torpedo level. Table 7 contains the detailed results of the tests. The final sensitivity is 8.799 N/V. The data collected was saved under the filenames and directories listed in Table 7.

Directory:	ham.8-11	ham.8-11	ham.8-12	ham.8-12	ham.8-13	ham.8-13
	(1st set)	*(2nd set)*	*(1st set)*	*(2nd set)*	*(1st set)*	*(2nd set)*
DATE:	11-Aug	11-Aug	12-Aug	12-Aug	13-Aug	13-Aug
filename	**hammer sensitivity**	**hammer sensitivity**	**hammer sensitivity**	**hammer sensitivity**	**hammer sensitivity**	**hammer sensitivity**
cal1.mat	11.8831	11.9132	9.074	10.9091	10.0927	8.9854
cal2.mat	6.3833	8.274	7.2076	12.617	10.0394	8.529
ca3.mat	10.3508	9.5921	7.7944	8.1275	6.7495	9.2374
ca4.mat	8.2919	8.0415	8.5006	7.322	9.0817	7.8727
cal5.mat	8.7263	7.3988	6.7107	8.3501	8.1551	7.5154
cal6.mat	7.5518	8.443	7.3919	8.3673	10.3222	7.6479
cal7.mat	11.4564	8.3922	9.3233	9.7051	10.7361	8.3321
cal8.mat	10.2308	8.0092	7.6666	7.0874	7.8665	8.3767
cal9.mat	6.7486	11.2838	7.8964	8.6463	8.8167	11.1995
cal10.mat	9.6312	9.3567	9.2402	9.159	9.2012	6.117
ave:	9.125	9.070	8.081	9.029	9.106	8.381
std dev:	1.9008	1.4833	0.9100	1.6807	1.2503	1.3224
%	20.83%	16.35%	11.26%	18.61%	13.73%	15.78%

	overall
ave:	8.799
std dev:	1.4566
%	16.55%

Table 7. Impulse Hammer Calibration Test Results

3. Accelerometer Setup

The next step in preparing the NPS Space Truss for modal analysis is accelerometer placement and setup. Two models of Kistler accelerometers were used in the modal testing: models 8690C10 and 8690C50. Their only difference is in their sensitivities. Model 8690C10 has a 10 g (where g = 9.807 m/s^2) maximum range and model 8690C50 has a 50 g maximum range. There sensitivities differ dramatically, averaging from 100 mV/g to 495 mV/g per axis, for the C50 and C10 models respectively. It will be necessary to take this into account when placing the accelerometers on the truss. If we put the more sensitive accelerometer at the extreme ends of the truss, where the first mode shapes have their greatest energy, we will overload the accelerometer. However, we want to retain this sensitivity for the inner nodes, where

movement is more restricted and so we will need a more sensitive measuring device (i.e. the C10s).

As mentioned earlier, the preferred mounting adhesive for the Kistler model 8690C10 and C50 accelerometers is the supplied Petro Wax from Katt & Associates in Zoar, Ohio. [Ref. 8] Before applying the wax adhesive, clean the flat surface of the thumb screw with ordinary alcohol. After cleaning the surface, attach the accelerometer by simply applying a thin (~0.1 mm) layer of wax to the flat surface of the thumb screw and apply the accelerometer to this layer with firm finger pressure and an alternating twisting motion. Only after attaching the accelerometer, are we ready to connect the cable. Immediately after connecting the cable, use a "ziplock" type fastener to fasten the accelerometer cable to a truss strut. Ideally, we just want to fasten the cable to the truss about four to five inches from the accelerometer connection. This protects the accelerometer from falling and damaging itself should the wax adhesive fail. Figure 11 is a good picture of a mounted accelerometer. In this case, it is node 41, an impact point (note the offset, extra thumbscrew as the impact target).

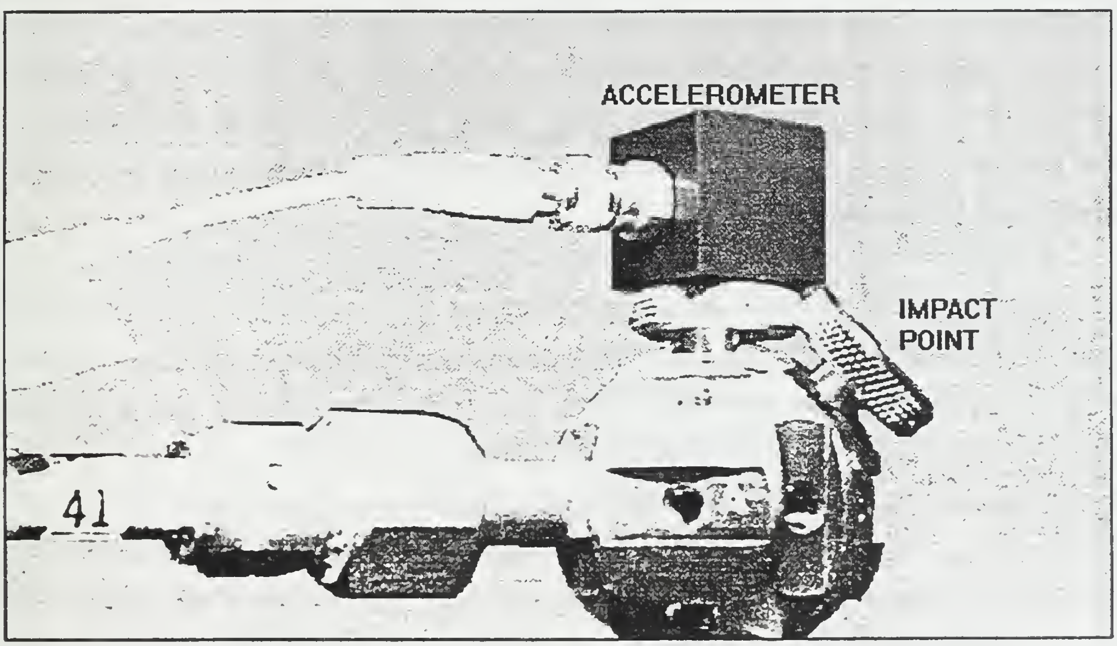


Figure 11. Accelerometer Placement (w/ Impact Point)

Now with the accelerometer in place, the cable connected and fastened to the truss, we want to level and align the accelerometer. The Newport table and NPS Space

Truss are already leveled to acceptable testing tolerances. Each accelerometer should have an ordinary, carpenter's torpedo level placed on all horizontal surfaces to check for level. Any adjustments necessary can be made by applying firm, finger pressure and a slight twisting motion (similar to the mounting technique) and then rechecking with the level until they are satisfactorily in place. Horizontal alignment (parallel with the long axis of the truss) is a more subjective alignment. Basically, one should stand so that their line of sight is along the line of accelerometers. By "eyeballing" the accelerometers in this fashion, a reasonable alignment can be made. When it is necessary to move (or remove) the accelerometers, simply twist and gently pull them off the thumb screws. Usually, a finite amount of wax is left on the mounting surface and the accelerometer. Sometimes it may not be necessary to reapply the wax adhesive to a new surface. This way, the remounting process is greatly expedited and the whole testing process takes less time.

It is up to the tester and the type of data being sought, that determines the accelerometer placement. To get a complete picture of the truss's response to input, every node should be tested. With eight accelerometers, this will require six separate tests in order to cover 48 nodes (excluding the four nodes fixed to the base plate). However, there are many other accelerometer placement scenarios to be explored. Several scenarios were used in the initial modal testing of the truss covered in this thesis. These placement scenarios will be covered briefly now, and in greater detail in Appendix D, Laboratory Experimental Test Log.

It should be noted that the accelerometer axis coordinates are different than the truss coordinates. Both the truss and the accelerometers have a conventional right handed coordinate system, however the electrical connectors on the accelerometers prevent mounting them in the same orientation as the truss. It will be import to remember this while analyzing the data. The actual coordinate systems are:

Accelerometer	Truss
+ x	+ x
+ y	- z
+ z	+ y

The global orientation between the truss coordinates, accelerometer coordinates, and the output channels to dSpace is in Table 8. The cable number, which identifies a signal conditioner input cable (eight input cables for eight accelerometers), is attached to

a specific accelerometer for each test. Their relationship is documented in Appendix D, the Laboratory Experimental Test Log.

cable	channel #	truss axis	accel. axis
1	1	x	x
	2	-z	y
	3	y	z
2	4	x	x
	5	-z	y
	6	y	z
3	7	x	x
	8	-z	y
	9	y	z
4	10	x	x
	11	-z	y
	12	y	z
5	13	x	x
	14	-z	y
	15	y	z
6	16	x	x
	17	-z	y
	18	y	z
7	19	x	x
	20	-z	y
	21	y	z
8	22	x	x
	23	-z	y
	24	y	z
n/a	25	(hammer)	

Table 8. Accelerometer – Truss Alignment

After conducting a complete nodal test (48 nodes over six tests), a separate test was conducted where all eight accelerometers were places in a line, from node 45 to node 52. This test setup was devised to capture an "image" of how the impact force travels down the length of the truss. Another test gave us a maximum spread of the accelerometers. All eight accelerometers were spread out along the length of the truss, such that both ends, and all longeron positions were covered. This test would give us a global image, in one impact, thereby bypassing any error induced when the

accelerometers were moved and separate impacts were used to excite them. Obviously, there are many other noteworthy tests worth exploring and which could be possible subject matter for future student theses.

4. Electronics Setup

There are several pieces of electronic equipment used during modal testing at the NPS Dynamics Laboratory. Specifically, accelerometers are connected to the Kistler multi-channel couplers (or signal conditioners), model 5124A (12 channels). These signal conditioners include a current regulator, buffer amplifier and decoupling network that removes the DC bias and passes the dynamic signal to the output, for each channel. [Ref. 10] Basically, the signal from the accelerometer passes along a cable (made by Kistler specifically for their accelerometers) to the signal conditioner, where it is amplified and split into three separate, independent signals for the x, y and z axis. In fact, each accelerometer connection cable has four wires inside: x-axis signal, y-axis signal, z-axis signal, and the ground. The signal conditioners are connected to dSpace, the Analog to Digital (AD) data acquisition system, via ordinary coaxial cables (one per axis, or three per accelerometer). The impulse hammer is connected to its own signal conditioner which is, in turn, fed into dSpace as well. Ultimately, twenty-five channels are fed into dSpace (24 from the accelerometers and one from the impulse hammer). Finally, dSpace is connected to a common, desktop PC for the final analysis. Software provided with dSpace, and modified for this specific testing displays the collected data. This same software saves the data in *.mat* files for further analysis by MATLAB.

B. COLLECTING DATA

A total of 90 tests were conducted during the course of modal testing of the NPS Space Truss. The data files from these tests, saved as *.mat* files, are located in the *c:\andberg\truss1* directory on the dSpace interface computer (desktop PC) in the NPS Dynamics Laboratory. A detailed list of filenames and dates on which the testing was conducted is located in Appendix D, Laboratory Experimental Test Log. Each test is a collection of data over 25 channels. Twenty-four channels (2 x 12) are the Kistler, multi-channel, signal conditioners connected to the Kistler, tri-axial accelerometers. The 25th channel is the PCB® Piezotronics, single channel, signal conditioner connected to the impulse hammer.

dSpace was configured for a 10 kHz sampling frequency taken over 0.5 seconds. For each test, the dSpace software was initiated, but data was not recorded until a predetermined trigger level was attained. For modal testing of the NPS Space Truss, the trigger level (measured by the impulse hammer) was set to 0.2 mV. This would filter out any weak, hammer impacts, and keep the computer from recording data until after the truss had been excited.

After turning on the signal conditioners and making sure that the accelerometers are securely connected to them, each channel's line must be tested by pressing the Front Panel Line-Test button. A green LED indicates a good condition (closed connection) whereas a red LED indicates that either the cable is damaged, there is an bad connection, and/or the accelerometer itself is damaged. Once the truss has been adequately prepared by the placement of the accelerometers, charging the damping pistons on the Newport table, turning on all equipment, testing the connections, and initiating dSpace and the host PC, data can be collected. To initiate dSpace, turn-on the dSpace host PC and turn on dSpace, invoke MATLAB and change directories to *c:/andberg/dspace.* At the MATLAB prompt type *mode.* When the SIMULINK block diagram appears, go to the "code" option and select "generate real time" (see Figure 12). This will open a DOS window, generate the code, and report download succeed, unless there is an error. At this point, close the DOS window, minimize the SIMULINK window, and execute the program *trace_40w.exe*, whose icon is in the Microsoft Toolbar.

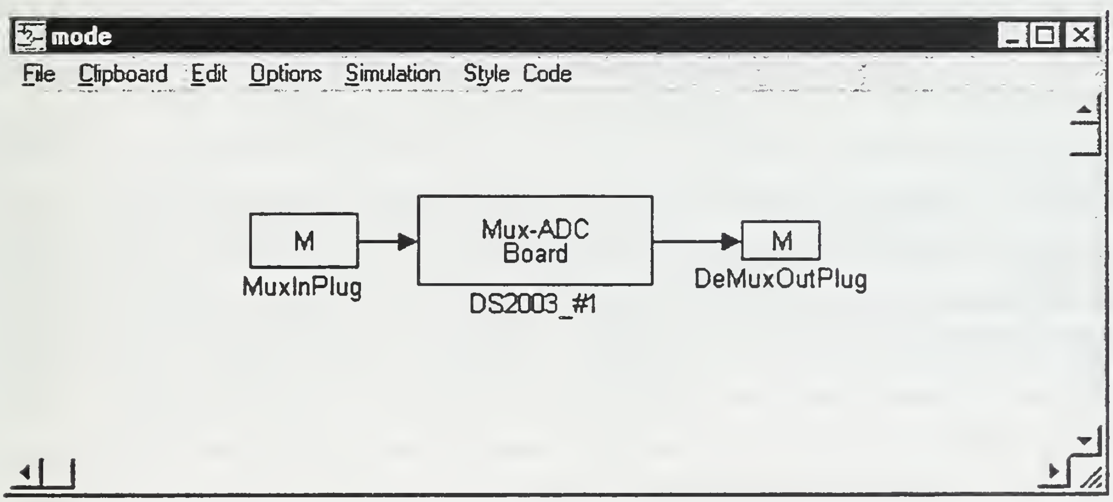


Figure 12. SIMULINK Window: *mode.m*

To begin collecting data, the impulse hammer should strike an impact point on the truss. It is important that the impact vector be equally distributed along all three axis of the truss (see Figure 13).

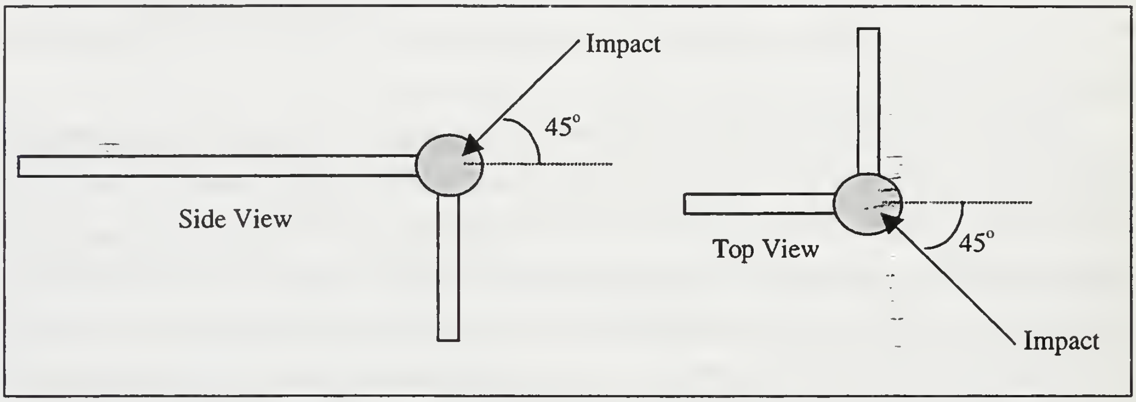


Figure 13. Impulse Hammer Impact Alignment

By collecting data over several trials, a reasonable average can be generated that will minimize the possibility of any bias towards a particular axis. Since the impulse hammer output will be used as the input to our simple relationship (recall: RESPONSE = PROPERTIES x INPUT), we will need to extract the x, y, and z components from the overall output. These simple coordinate transformations into the truss coordinate system are express in the following equations:

$$\text{x axis input value} = \cos^2(45^\circ) \qquad \textbf{(3.3a)}$$

$$\text{y axis input value} = \sin(45^\circ) \qquad \textbf{(3.3b)}$$

$$\text{z axis input value} = \cos(45^\circ) \times \sin(45^\circ) \qquad \textbf{(3.3c)}$$

The truss's x-axis and z-axis are scaled the by the same amount (scale factor of 0.5). The truss's y-axis, however, is scaled less then the others (scale factor of 0.707). Each impact with the hammer should produce a clean spike displayed by the dSpace software. Impacts ranged anywhere from 250 mV to 700 mV. Usually, the more energy in the impact (higher mV response level), the better the data collected. The only drawback to an energetic impact is that the more sensitive accelerometers may become saturated (or overloaded). Should this be the case, a softer impact may be desired. The same individual should apply the impact hammer for each test for consistency.

After each test (or trial), it is important to damp the truss simply by gently holding a strut for a few seconds. Also, a periodic check (ever few tests) of the accelerometers' alignments with the carpenter's level will prevent the possibility of corrupted data from a misaligned accelerometer.

C. ANALYZING DATA

1. Global Analysis

The first step in analyzing the NPS Space Truss modal data was to develop MATLAB code which allowed the user to view multiple plots of accelerometer data, after scaling them by their inherent sensitivities, as well as the impulse hammer impact scaling. This code is called *xfer.m* and can be viewed in Appendix G, *xfer.m* – MATLAB Analysis Code. *xfer.m* uses the *tfe.m* (Transfer function estimate) function available in the MATLAB Signal Processing Toolbox. [Ref. 11] The *xfer.m* code will find a transfer function estimate T_{xy} given an input signal vector x and output signal vector y. The resulting transfer function, which is the quotient of the cross spectrum of x and y and the power spectrum of x, is given in equation (3.4): [Ref. 11: p. 2-218]

$$T_{xy}(f) = \frac{P_{xy}(f)}{P_{xx}(f)} \tag{3.4}$$

In our transfer function, the input is taken from channel 25, the impulse hammer, and the output comes from channels one through 24, or the accelerometers.

The data recorded as *test78a.mat* was analyzed first (for all filenames, see Appendix D, Laboratory Experimental Test Log, for details). This test gathered data from all eight accelerometers being set up in a line, from node 45 to node 52. Data was gathered for 0.5 second at a 10 kHz sampling frequency over all 25 channels. Figure 14 displays channels 1, 13, 10, 22, and 16, which corresponds to the truss x-axis: nodes 52, 50, 48, 46, and 45 respectively. The solid, vertical, green lines are the computed natural frequencies superimposed over the plot as a reference.

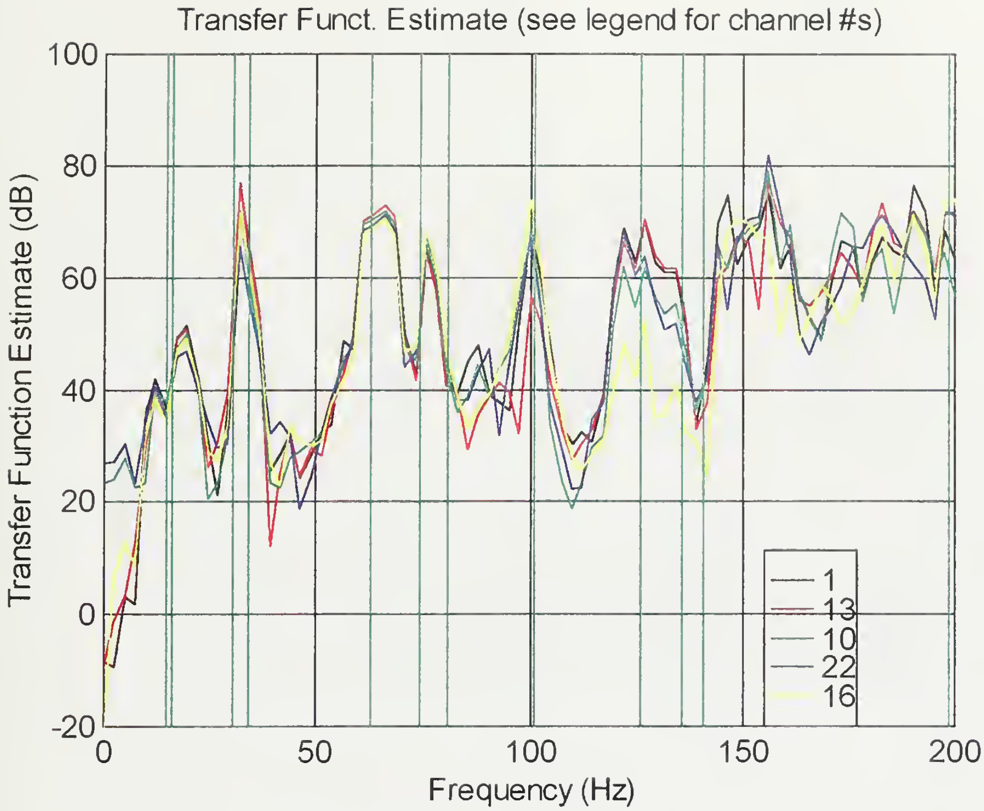


Figure 14. Plot of *test78a.mat* (x-axis)

There is a good deal of energy (≥ 70 dB) concentrated around 33 Hz, 65 Hz, 75 Hz, 100 Hz, 124 Hz, 130 Hz, 155 Hz, and so on. It is interesting to note that these energy concentrations are consistent for each channel. Also, this spread of channels provides a broad representation of the nodes sampled in the test.

Comparing this data (truss x-axis only) to the same nodes, same test, but now the z-axis (truss), we notice that the energy concentrations are generally at the same frequencies, but in different magnitudes. Specifically, the energy concentrated at 100 Hz is on the average a good 20 dB greater along the truss z-axis (see Figure 15). Again, solid green lines representing the computed natural frequencies are superimposed over the measured data as a reference.

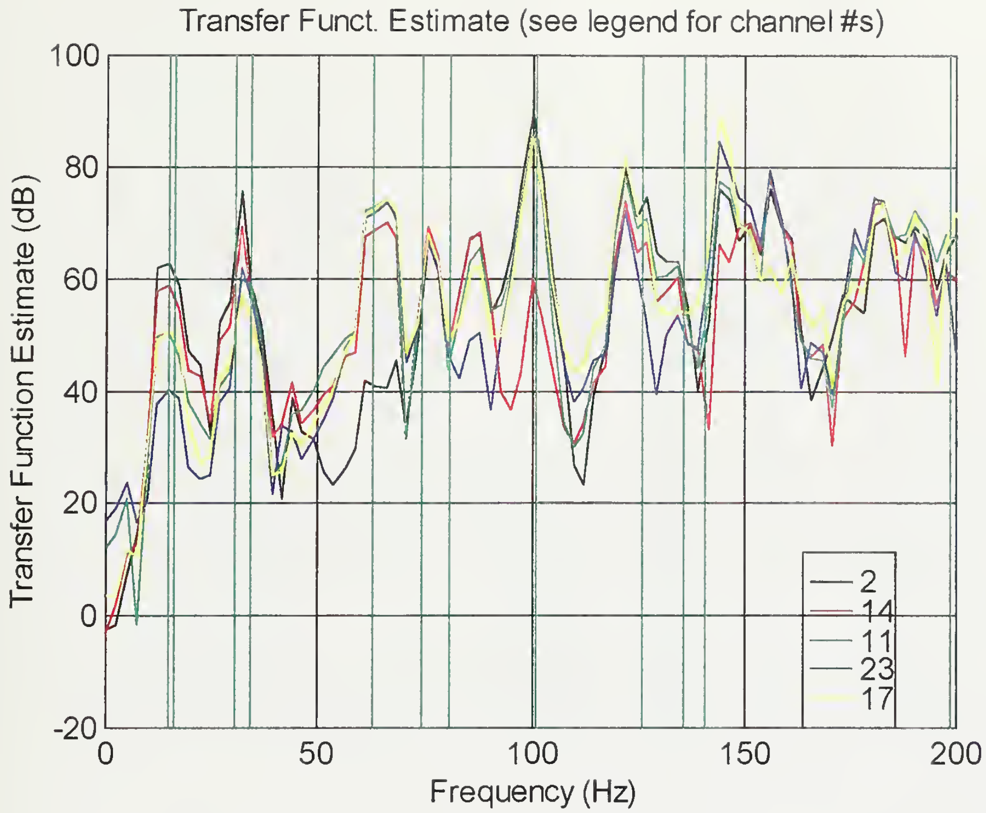


Figure 15. Plot of *test78a.mat* (z-axis)

Additionally, channel 17 (corresponds to truss z-axis, node 45) begins to dominate at the higher frequencies (see Figure 16). At approximately 142 Hz and around 275-280 Hz, channel 17 clearly dominates over the other channels. We will recall that the truss has greater freedom to move in the truss y and z directions. Also, for *test78a.mat*, node 41 was the impact point (see Figure 9, Impact Node Locations, for an idea of the proximity of the impact point and node in question). These two facts provide a good explanation for this observation.

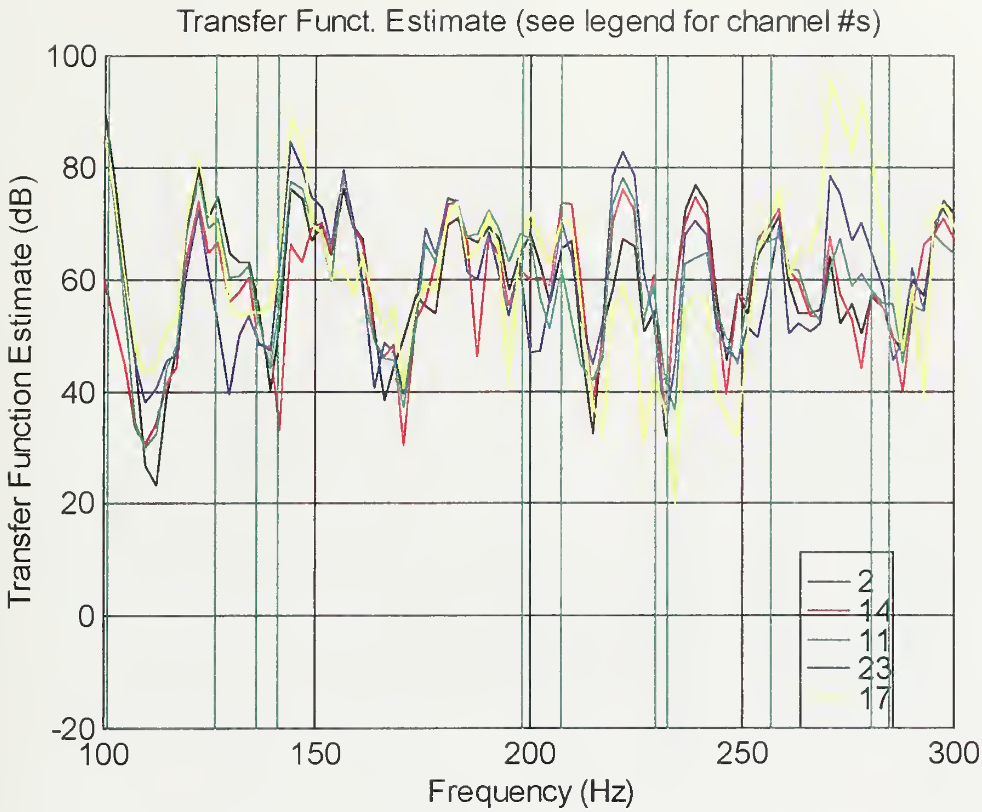


Figure 16. Plot of *test78a.mat* (z-axis), Higher Frequencies

The next global analysis of the truss will look at the data in file *test84a.mat*. Accelerometers were placed on nodes 3, 44, 14, 20, 52, 49, 41, and 11. In Figure 17, we will be looking specifically at channels 15, 21, 9, and 18 which correspond to the truss y-axis at nodes 3, 44, 49, and 14 respectively. This spread includes nodes at both extreme ends, and in the middle of the truss.

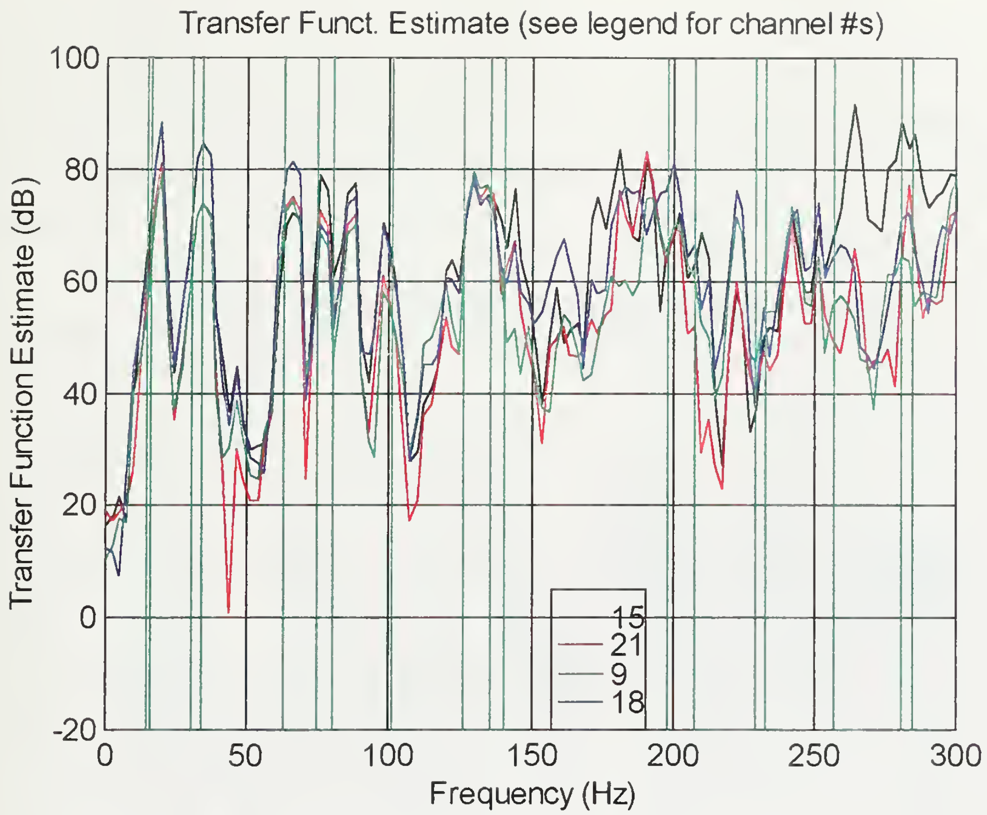


Figure 17. Plot of *test84a.mat* (y-axis)

Again, we note which channels dominate the plot. Channel 18 (node 14, truss y-axis) dominates the lower frequencies (from 25 to 75 Hz) while channel 15 (node 3, truss y-axis) dominates the higher frequencies (from 250 to 300 Hz). Recall that the *–a.mat* suffix on the filename means that node 41 was the impact point for this test. Channel 15 (node 3) probably gets most of its energy (up to 90 dB at 260 Hz) from its proximity to the impact point (see Figure 8, Impact Node Locations). Channel 18 and channel 15 also measure a great deal of energy (acceleration) because they are both on the extreme (and opposite) ends of the truss, where the truss's movement is least restricted.

Figure 18 displays that same nodes and the same axis but uses the data from file *test87b.mat*. In this case we are using node 24 as the impact point. As such, all nodes register less energy (only once, at 185 Hz, does the plot break 80 dB) then the nodes plotted in Figure 17. Node 24 is close to the center of the truss, where it is most rigid. Comparing this data with Figure 19, however, we will notice that the truss x-axis contains a noticeable amount of energy. In Figure 19, the same nodes are being analyzed as in Figure 18, but with the truss's x-axis data displayed. Channel 7 (node 49) dominates at

275 Hz. Node 49 is in close proximity to the impact point, node 24. Channel 16 (node 14) is also prominent around 180 Hz and 280 Hz. Node 14 is close to node 24 (impact point). Both of these nodes (49 and 14) get there energy from their proximity to the impact point. Once again, the green, vertical lines are references for the computed natural frequencies.

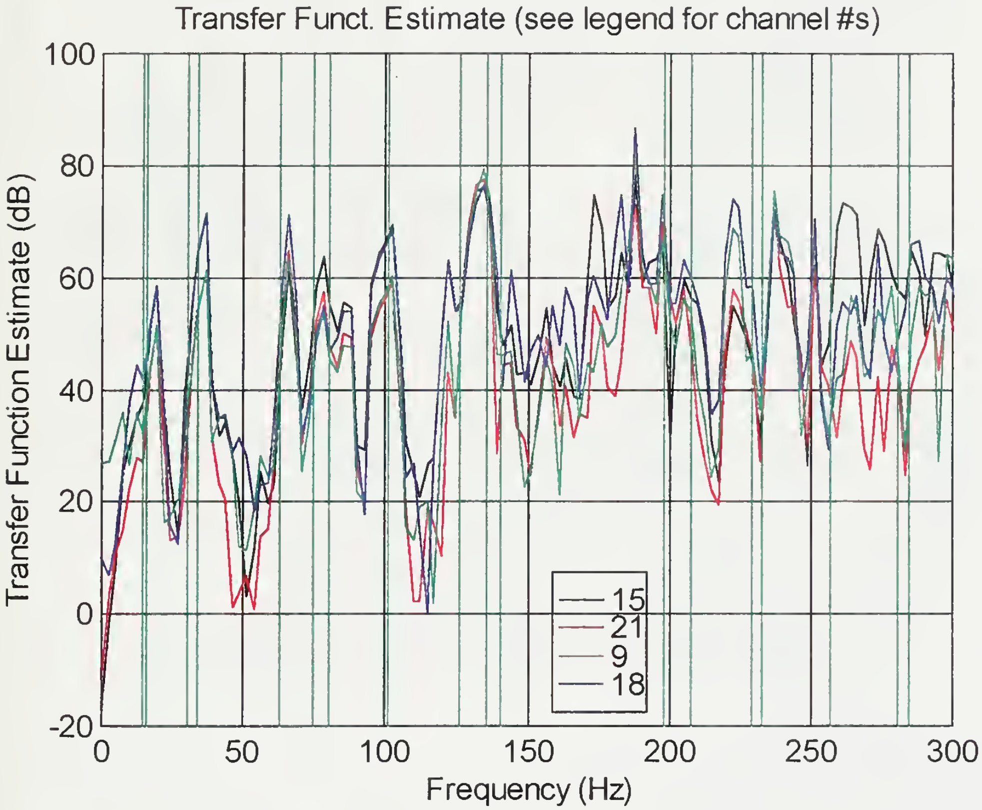


Figure 18. Plot of *test87b.mat* (y-axis)

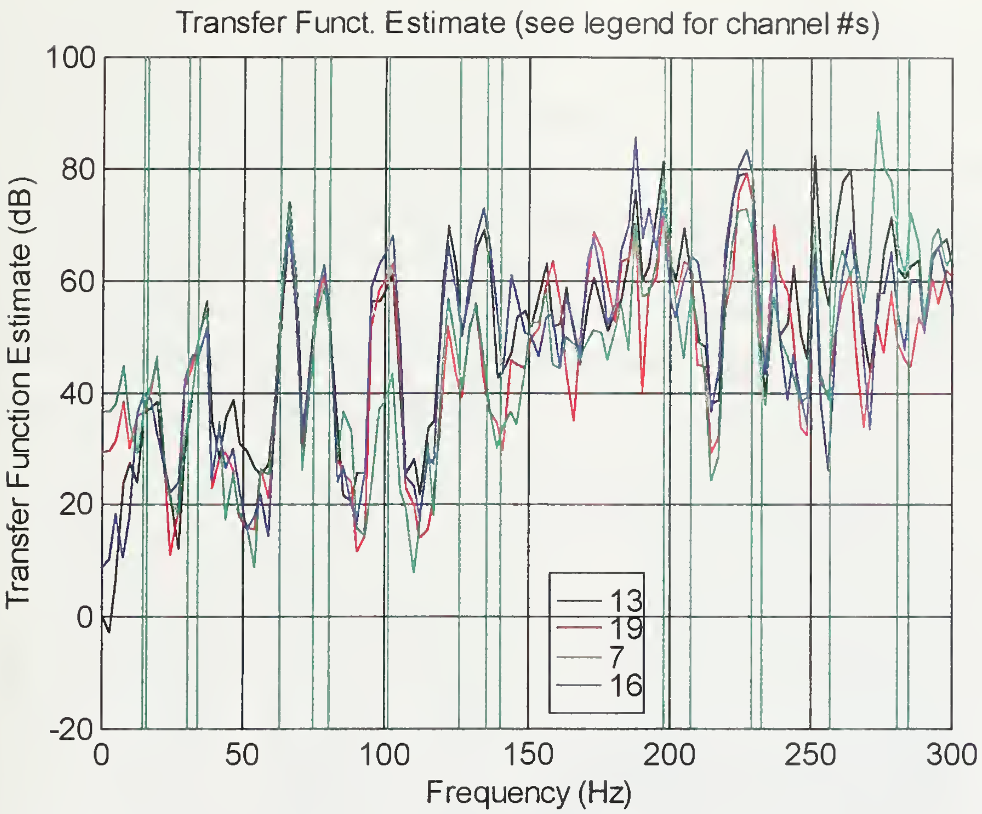


Figure 19. Plot of *test87b.mat* (x-axis)

2. Correlation of Experimental Data with Analytical Data

Figure 20 and 21 displays the plots of eight different channels from the data in file *test16a.mat*. This data was collected at the extreme end (eight nodes) of the truss where the acceleration measured will be the highest. Node 41 was the impact point and sampling was done at 10,000 Hz over 0.5 second. Figure 20 displays data collected from the truss y-axis and Figure 21 displays data collected from the truss z-axis. Both of these plots have the NPS Space Truss natural frequencies (0 Hz – 300 Hz) displays over them as solid, vertical black lines. Referring back to Table 4, NPS Space Truss Natural Frequencies (calculated), we can correlate the computed natural frequencies with the measured natural frequencies (these values are the same as the vertical green lines). In both plots (20 & 21), the natural frequencies from 0 Hz to 100 Hz match almost exactly. We will notice that the predicted natural frequencies: 14.64 Hz and 16.26 Hz, appear to be combined around 16 Hz in our plots. This may be due to a lack of resolution in the

collected data. We are sampling at 10,000 Hz over 0.5 second for each channel. A higher sampling frequency may yield a finer resolution. However, this also requires significant more computer memory to store all these data points in a real-time environment.

Beyond 100 Hz, however, the collected data does not correlate as nicely with the computed data. We still see approximate matches, specifically around 126 Hz and 135 Hz, as well as around 230 Hz and 280 Hz. At these higher frequencies, we may be measuring the first few natural frequencies of the individual struts (as opposed to the natural frequencies of the entire truss). Equally as likely, the FEM code may just do a poor job of predicting the higher frequencies.

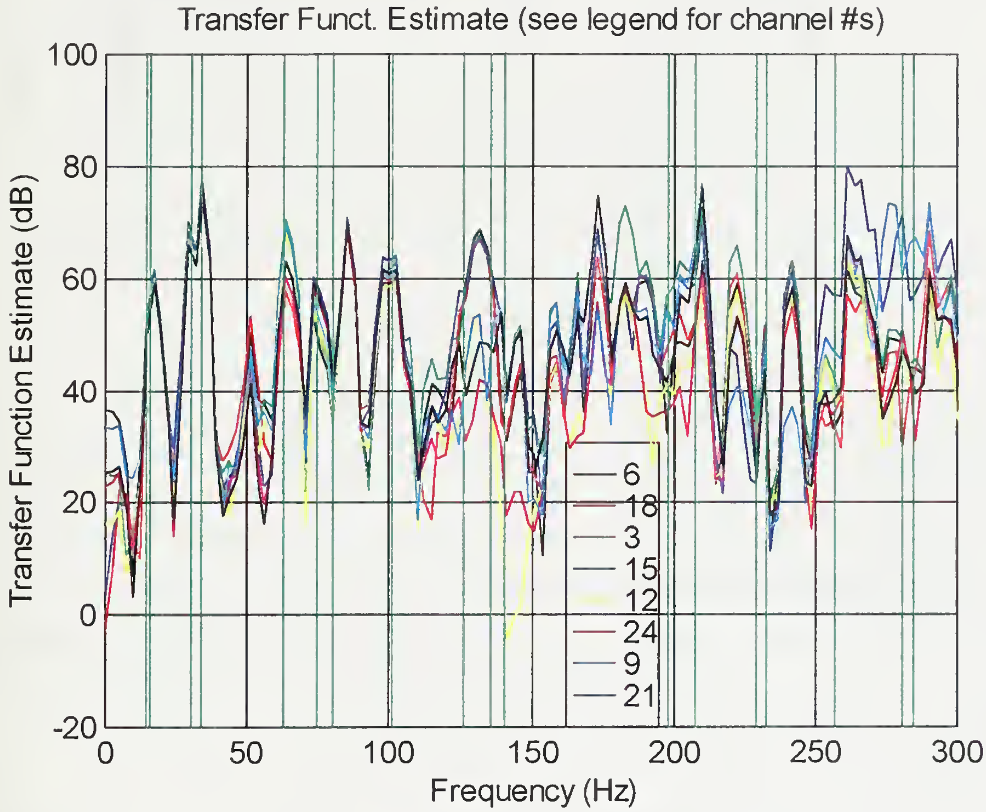


Figure 20. Plot of *test16a.mat* (y-axis)

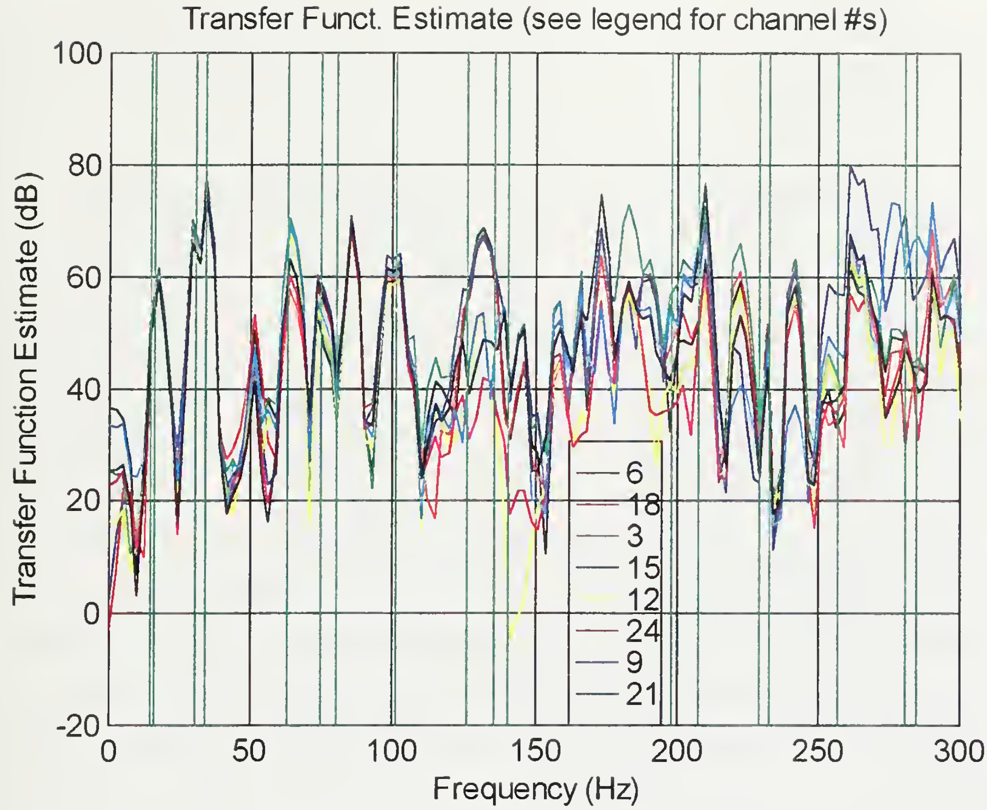


Figure 21. Plot of *test16a.mat* (z-axis)

In this next analysis, we will be looking at data from files *test38a.mat*, and *test45b.mat*, and plotting their measurements from the y-axis and z-axis, respectively. Both of these file contain data acquired near the center of the truss. We are specifically interested in nodes 7 and 19. There is relatively a fair amount of energy located at these two nodes. In addition, we will compare our measurements to the NRLFEMI function: Calculate FRF (see Appendix A, NRLFEMI User Instructions).

The NRLFEMI, Calculate FRF function requires several user inputs. For the data in Figure 22, We will command NRLFEMI to take 100 samples over one second. The excitation node was 24, in the z direction and the response was taken from node 7, in the z direction (this simulation will be the closest to our test case: *test45b.mat*).

Figure 22 plots data from *test45b.mat* along with the matching NRLFEMI plot of the same input and response node (27 and 7 respectively) for the same frequency range. Once we superimpose the computed natural frequencies over both plots in Figure 22, we see how well the model matches the collected data. In this Compute FRF case, we are only getting input (energy) along the z-axis. However, in the test case, we are exciting the

truss equally along all three axes. This may explain the difference in magnitudes between the two plots. One aspect to keep in mind is that we are now observing the more rigid modes.

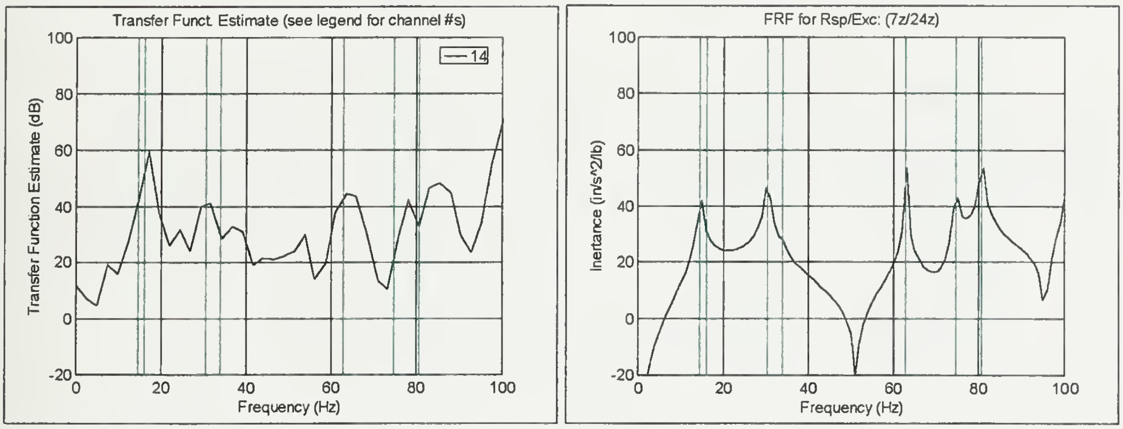


Figure 22. Plot of *test45b.mat* (z-axis) with NRLFEMI Plot of Same Response

In Figure 23, we are looking at data from *test38a.mat* along side of the NRLFEMI, Calculate FRF plot. As previously mentioned, node 41 is the excitation node for test38a.mat, and we are interested in node 19 for the response. We will command NRLFEMI to take 100 samples over one second. In this case, the excitation energy, as well as the response, will all be along the y-axis. Superimposing the computed natural frequencies over the plotted data, we can correlate the measured data with the computed data.

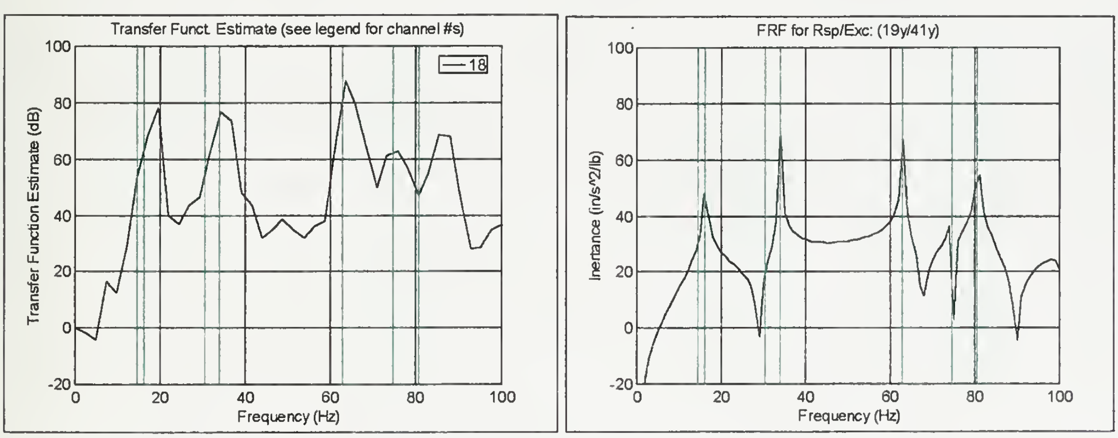


Figure 23. Plot of *test38a.mat* (y-axis) with NRLFEMI Plot of Same Response

Notice in Figure 23, the first computed natural frequency (14.64 Hz) does not stand out in either plot. The fourth natural frequency (33.97 Hz) seems to dominate over the third (30.41 Hz), in both plots as well. As in Figure 22, the energy measured in Figure 23's experimental data is greater then in the computed data. This may be because the input for the experimental is along all three axes, whereas we commanded NRLFEMI to take input only from the y-axis.

The four figures just referenced (Figures 20, 21, 22, & 23) provide good support for NRLFEMI's calculated natural frequencies, especially for the first ten to thirteen natural frequencies. Other figures using more of the collected data will be available in Appendix I, Data Plots.

IV. FIBER BRAGG GRATING SENSORS

A. INTRODUCTION TO FIBER BRAGG GRATINGS

1. Description of Fiber Bragg Gratings

"Fiber Bragg gratings (FBG) reflect a specific wavelength that shifts slightly depending on the strain applied to the FBG sensor." [Ref. 4: p. 2] In other words, a Fiber Bragg Grating Sensor (FBGS) is a type of optical strain gage. Additional FBGS applications include: long-term static strain sensing (health monitoring), dynamic strain sensing, temperature and pressure sensing, magnetic and electric field sensing, and chemical sensing. There are several unique characteristics separating FBGSs from other, conventional sensors. Large numbers of FBGSs can be placed at predetermined locations, all co-located on the same strand of fiber. This string of quasi-distributed, quasi-point sensors can be interrogated simultaneously using wave division multiplexing or time division multiplexing. Additionally, due to the material and geometry characteristics of the fiber itself, FBGSs can be embedded in composite smart structures.

After briefly discussing the many applications of FBGSs, a physical description and discussion of their fabrication is given. Optical fibers are made of fused silica, an ideal, high-temperature elastic material. Actually, silica filaments may have ultimate tensile strengths on the order of 1 Mpsi for a small segment. [Ref. 5] This translates into a fiber being able to withstand strain approaching 8%. Fibers used in FBGSs can potentially tolerate a higher strain-to-failure than most of the reinforcing filaments used to construct high-performance composites. Hence any sensor embedded in a composite should live as long as the structure and not induce premature failure. After drawing the silica fiber core, a coating is applied. In quality fibers, this coating is a high-temperature/high-modulus polymer, such as a polyimide. This coating increases the overall diameter of the fiber by only 10 μm. This polyimide coating provides good adhesion to epoxies, important when attaching FBGSs to a structure, while not degrading during the epoxy curing process.

To create a FBGS, the polyimide coating is stripped away over the length of the sensor. Then, using two interfering, high energy UV beams (holographic), a grating is formed by laterally exposing the stripped length of fiber to a three-dimensional fringe

pattern. [Ref. 6] The interaction of the UV beams with the silica fiber creates a permanent modulation of the refractive index of the fiber core. Figure 24 is a visual representation of the fabrication of FBGSs. Repeating this procedure produces a distribution, or pattern of gratings that comprise the FBGS.

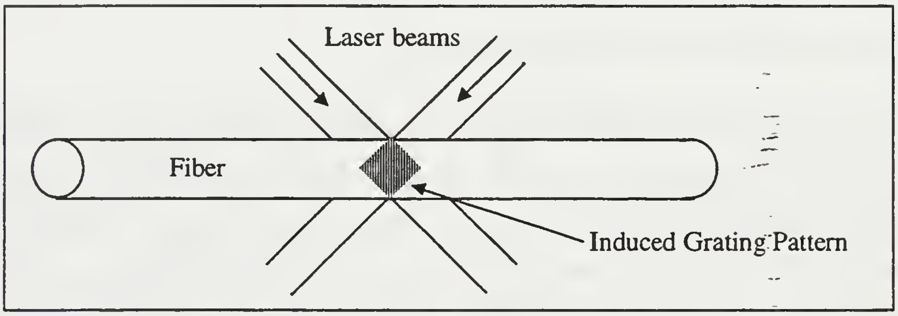


Figure 24. FBGS Fabrication

The pattern length is programmable anywhere from 1 mm to several centimeters (the Bragg Photonics FBGSs that NPS employs are 10 mm long). The modulation pattern can be adjusted to the user's needs for a wide range of wavelengths. Industry standards define the working wavelength range for FBGSs at 1290 to 1330 nm and 1520 to 1570 nm, however the modulation of the refractive index can range anywhere from 400 nm to 2000 nm with present technology. A FBGS is, in essence, a tuned optical filter. The FBG will pass all wavelengths with negligible to no attenuation while filtering a predetermined wavelength and reflecting it back down the fiber. The bandwidth of a typical filter is on the order of 0.1 nm in the 1330 nm band. This is with a transmission loss (or reflection) of up to 100% (ideal). [Ref. 5]

FBGSs return a measurement in the form of a wavelength shift in their back-reflected spectrum. The filter's wavelength is set to return a very narrow spike at its predetermined Bragg wavelength. Figure 25 shows the Bragg, reflected wavelength for a typical FBGS. In this case, the returned signal is a narrow bandwidth of light (FWHM: 0.290 nm) centered at 1550.090 nm. This spike will shift when the FBGS is placed under tension or compression.

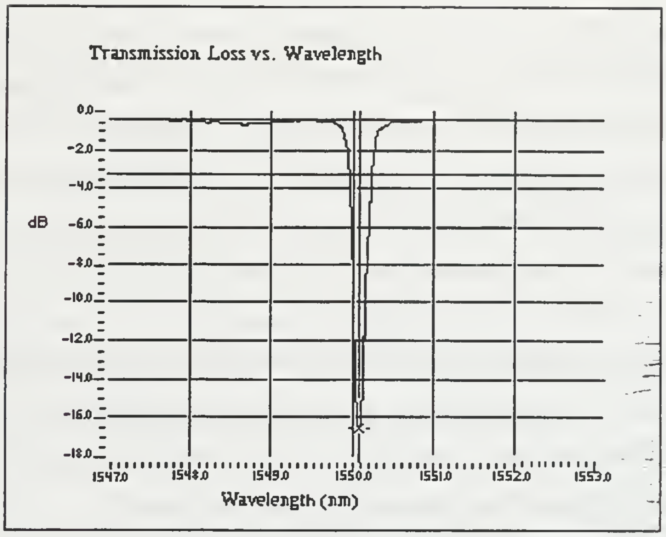


Figure 25. Transmission Loss vs. Wavelength for FBGS

This Bragg wavelength, λ_B, can be related to the effective core index of refraction, n, and the period of the index of modulation, Λ, by the following equation:

$$\lambda_B = 2n\Lambda \tag{4.1}$$

If the fiber grating is strained, the wavelength of the reflected light changes slightly such that:

$$\delta\lambda_B / \lambda_B = -0.74\delta\varepsilon \tag{4.2}$$

where ε is strain.

The ability to predetermine the filter's wavelength is important when employing multiple FBGSs on one strand of fiber. Usually, a FBGS's wavelength is recorded during manufacture. When several FBGSs are connected in series, they are usually arranged in increasing (or decreasing) wavelengths. In this fashion, it is easy to determine which wavelength corresponds to which measurement a FBG interrogation system is returning. As long as their position on a structure is properly documented (i.e. which FBGS wavelength corresponds to which element), a real-time picture of the structure's strain is readily available.

There are some very desirable characteristics of the Bragg gratings in a FBGS. First, the linear response of the center wavelength prevents any direction ambiguity

during measurements. When the grating region of the fiber is disturbed in a manner that modifies the distributions of the gratings, the index of refraction changes accordingly. A small (non-permanently deforming) change along the longitudinal axis of the FBGS causes the narrow spike, or peak, in the returned wavelength, to shift to longer wavelengths during tension and shorter wavelengths during compression. Disturbances in the grating can be due to mechanical strain, temperature change, or any other physically altering condition. Second, absolute measurement is possible since the Bragg grating center wavelength determines the measured value in terms of a reference state. Third, the grating tends to reject the effect of strain fields not aligned with the longitudinal axis. In fact, the unidirectional, transverse components are attenuated by a factor of 500 or more compared with the longitudinal response.

FBGSs are ideally suited for real-time evaluation of load, strain, vibration, and other health monitoring functions of structures. A string of FBGSs on a single strand of fiber in a multiplexed network can provide light weight, highly durable, low power, and accurate feedback for use in controlling smart structures. The application of FBGSs to the NPS Space Truss provides a unique ability to compare conventional accelerometer readings to the dynamic feedback from a FBGS. Eventually, this feedback will be incorporated in a control law with actuators, transforming the Space Truss into a smart structure.

2. Application of Fiber Bragg Gratings

The greatest potential for compromised FBGS readings lies in their application. A weak application, or one that significantly attenuates vibration, will fail to properly and accurately transfer the physical changes (strain, etc.) of the structure being measured to the FBGS. In other words, if an adhesive is incapable of transmitting high frequency vibrations for example, the FBGS will never measure those vibrations or pass that information on to the FBGS interrogation system. This said, it is very important to choose a suitable adhesive, prepare the surface to receive the FBGS, and carefully apply the FBGS to the structure in question.

The first in applying FBGSs was to choose a suitable structure for the technology demonstration. A plate of aluminum was found and cut into a plank. This plank was then fastened onto a mounted vice, making a simple beam. Next, the surface of the beam was prepared to receive the adhesive and the fiber.

- CSM-1A Degreaser is the preferred chemical for degreasing a surface prior to FBGS application. As a substitute, however, Isopropyl Alcohol can also be used. The end result should be a surface free from any contaminants, especially oil from hands, etc.
- Mark the surface with a 4H (hard) drafting pencil where the FBGS is to be placed. This will help by guiding the placement of the fiber onto the prepared surface.
- *M-Prep* Neutralizer 5A should be liberally applied to the application area. Keeping the surface wet, scrub with cotton tipped applicators. Do not allow evaporation of the cleaning material of the specimen surface since this would leave a thin, unwanted film between the adhesive and the specimen. Remove the Neutralizer by slowly wiping through the gage area, allowing the gauze sponge to absorb the Neutralizer. Do not wipe back and forth over the gage area since this may allow contaminants to be redeposited on the cleaned area.
- Carefully remove the FBGS from its shipping case and gently remove the plastic, protective sheath (this may already be off if several fibers were spliced together). Using two small (4 inches) pieces of clear, plastic tape, lift the FBGS up and have it ready to place down on the application area. Place the tape on either side of the actual FBGS, far enough away that they don't get in the way of the bonding agent. Eventually, the tape will hold the fiber in place while the bonding agent cures.
- Apply a small amount of the M-Bond Adhesive to the back of the FBGS and apply some to the application area. Immediately after this, place the fiber onto the structure, carefully aligning the fiber with the pencil marks made earlier. Also, slight tension should be applied so that the fiber is as straight as possible when the agent cures. This is where the plastic tape is convenient. The tape can maintain the tension in the fiber while the agent cures.
- After aligning and placing the fiber, apply a liberal amount of the M-Bond Adhesive over the FBGS and the application area. Once it is thoroughly covered in adhesive, allow at least an hour for the bonding agent to cure at room temperature.
- NOTE: The bonding procedure is irreversible. Once bonded, a fiber will be destroyed if it is removed. Make sure that all necessary calculations and

planning have take place prior to applying FBGS to a structure. Avoid allowing the M-Bond Adhesive to come into contact with unprotected skin.

The greatest potential for erroneous or false data when taking FGBS readings arises in the application of the sensor itself. The FBGSs are relatively fool-proof. Due to their linear feedback, and increased attenuation in the off-axis transverse components, FBGSs provide very reliable data. However, if the data that they are measuring (strain, vibration, etc.) is not accurately transmitted through the bonding agent, the data that the fibers return is useless. Therefore, it is imperative that a bonding agent with a completely linear transmission and minimum attenuation of the transmitted information be used in the application of FBGSs.

3. NPS Fiber Bragg Grating equipment

The Naval Postgraduate School's dynamics laboratory employs BRAGG Photonics FBGSs connected to a Micron Optics Inc., *picoWave*™ Fiber Bragg Grating Interrogation System (FBG-IS Serial Number 3005). This interrogation system can effectively monitor up to 31 FBGSs multiplexed on a single fiber. However, as the number of FBGSs increases on a fiber, their individual bandwidth decreases, compromising their overall dynamic range. Micron Optic's *picoWave*™ can resolve changes in the optical wavelengths as fine as 1pm (<1 μstrain) and achieve calibrated wavelength accuracy of ± 5 pm (4 μstrain). The following schematic (Figure 26) is representative of the *picoWave*™ interrogation system:

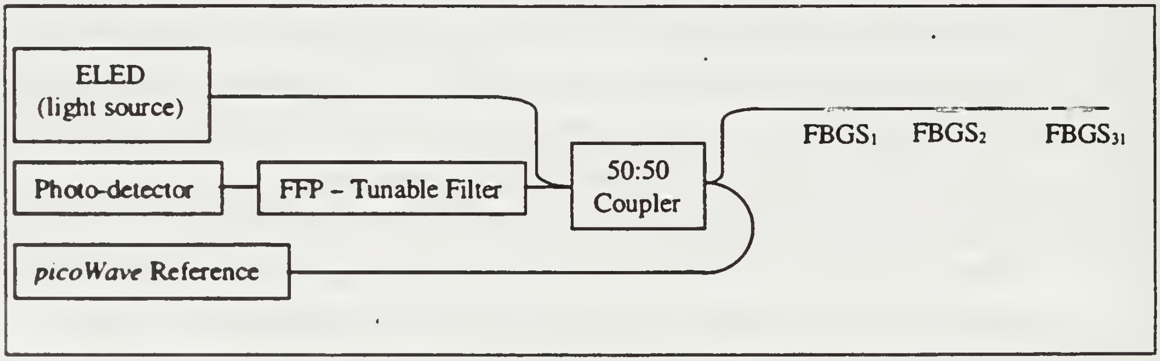


Figure 26. Schematic of *picoWave*™ Interrogation System

A drawback to the *picoWave* FBG-IS resides in its sampling frequency. Currently, the *picoWave* FBG-IS can perform dynamic strain sensing up to 25 Hz. [Ref. 4] Its sampling frequency is 50 Hz. To properly take dynamic measurements, specifically in regards to spacecraft sensing applications, much higher sampling frequencies are necessary. Micron Optics Inc. is planning to manufacture a FBG-IS with a higher sampling frequency then currently available, in the near future.

A simple, cantilevered beam affixed with a single FBGS (from Bragg Photonics) has been set up in the NPS dynamics lab. Figure 27 provides a good picture of the beam setup, next to the *picoWave* interrogation system.

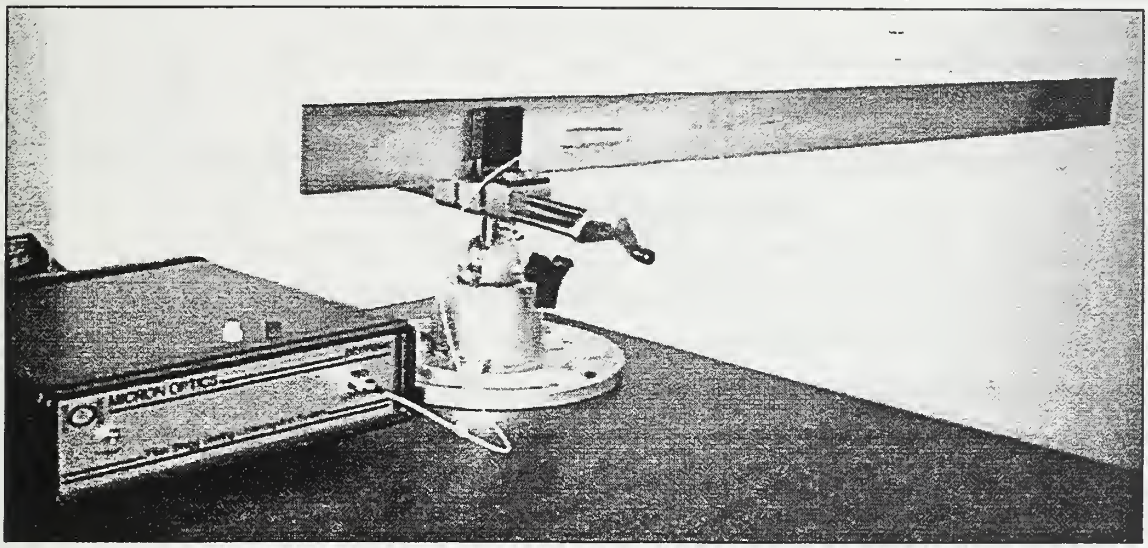


Figure 27. Fiber Bragg Grating – Interrogation System

The *picoWave* FBG-IS provides dynamic feedback, which is displayed on a desktop PC connected via an interface cable to the interrogator box. The displayed data is in the form of a time history vs. μ-strain plot (see Figure 28).

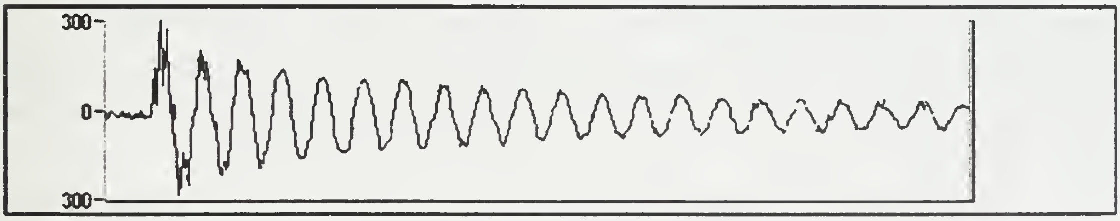


Figure 28. FBG-IS Display

This image was captured after exciting the 2nd natural frequency of a simple cantilevered beam. The 2nd natural frequency briefly "rides" on the first natural frequency for about the first two cycles. After that, only the 1st natural frequency remains as it slowly damps out. The scale on the left is in μ-strain.

One of the current applications of FBGSs is health monitoring of structures. To effectively do this, FBGSs will have to accurately measure a consistent strain over an extended period of time. Figure 29 is a representation of a FBGS measuring prolonged strain.

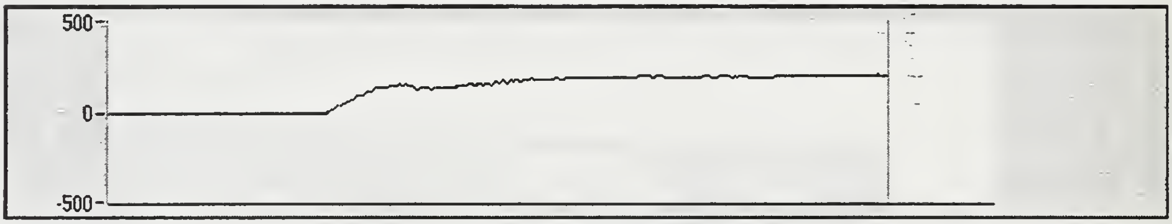


Figure 29. FBG-IS Display (strain)

V. CONCLUSIONS AND RECOMMENDATIONS

After a detailed analysis, the FEM model of the NPS Space Truss (NRLFEMI) closely predicted the measured natural frequencies from 0 Hz to about 100 Hz, and in some cases, even up to 208 Hz. The only exception was that the first natural frequency (14.64 Hz) predicted by the FEM (NRLFEMI) was not observed in the experimental data. A possible reason for this could be the proximity of the first and second natural frequencies, 14.64 Hz and 16.26 Hz, respectively. The second natural frequency may dominate over the first in magnitude. Magnitudes of the response at various natural frequencies were only briefly computed and deserve future attention. Additionally, analysis matched expected results when comparing data generated by the different impact points, and taken from different nodes along different axes of the truss. Detailed descriptions of the design, fabrication, and testing of the truss contained in this thesis should provide valuable information to follow-on research.

The NPS Space Truss continues to provide many excellent student opportunities to further research in the areas of modal testing and analysis, application of FBGSs, and development of control laws and implementation techniques for smart structures. Specifically, the NRL has recently made available the ability for an NPS student to telnet to NRL's facility and use their X-Modal platform to perform quick fits of collected data (see Dr. Bosse in Appendix F, Important Points of Contact).

As covered earlier in this thesis, the NRLFEMI program used to predict the truss's natural frequencies has several limitations. A better FEM should be developed, or improvements made to the existing NRLFEMI, in order to make more accurate predictions. Eventually an FEM will have to be developed which accurately describes the mode shapes of the truss, and is adaptable to smart struts while implementing control laws sufficient to damp unwanted vibrations.

A great deal of follow-on work exists in the area of FBGSs. At this point, a simple technology demonstration along with a description of the NPS FBGS system has been completed. Application of FBGSs to the NPS Space Truss, and more advanced data acquisition and analysis routines need be developed. Eventually, there should be a program written that can take dynamic data from the FBGSs and feedback this data into a control routine, transforming the NPS Space Truss into a smart structure.

LIST OF REFERENCES

1. Devore, Jay L., *Probability and Statistics for Engineering and the Sciences*, Duxbury Press, Belmont, CA, 1995.

2. Hallauer, W. L. and Lamberson, S. E., "A Laboratory Planar Truss for Structural Dynamics Testing," *Experimental Techniques*, Vol. 13, No. 9, 1989, pp. 24-27.

3. Allen, David H. and Haisler, Walter E., *Introduction to Aerospace Structural Analysis*, John Wiley & Sons, New York, NY, 1985.

4. *Fiber Bragg Grating Interrogation System Instruction Manual*, Document # 3.0a, Micron Optics, Inc., Altanta, GA, 1997.

5. Udd, Eric, editor, *Fiber Optic Smart Structures*, John Wiley & Sons, Inc., New York, NY, 1995.

6. 3M (Minnesota Mining and Manufacturing), "Fiber Bragg Gratings Application Note," "The Mechanical and Optical Reliability of Fiber Bragg Gratings," February 1996.

7. Ewins, D. J., *Modal Testing: Theory and Practice*, Research Studies Press Ltd., John Wiley & Sons, New York, NY, 1984.

8. *Kistler Operating Instructions*, PiezoBEAM® Accelerometers, Kistler Instrument Corporation, Amherst, NY, 1997.

9. *Newport Vibration Control System Instruction Manual*, Newport Corp., Irvine, CA, 1991.

10. *Kistler Operating Instructions*, Model 5124A Multi-Channel Couplers, Kistler Instrument Corporation, Amherst, NY, April 1989.

11. Krauss, T.P., Shure, L., and Little, J.N., *MATLAB Signal Processing Toolbox,* The Math Works, Inc., Natick, MS, February, 1994.

APPENDIX A. NRLFEMI USER INSTRUCTIONS

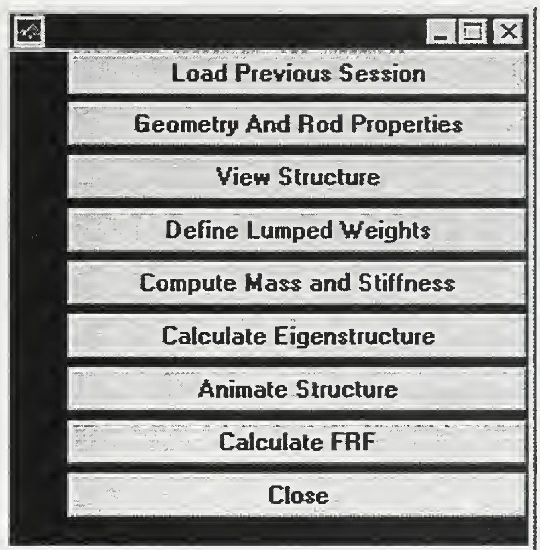
This appendix will provide the user step by step instructions on setting up their own truss properties (or modifying existing ones), and generating pertinent data.

Step 1. Start Windows on select PCs in the NPS Dynamics Lab.

Step 2. Run MATLAB

Step 3. Make sure that you are in the *c:\matlab\truss\fem* directory. You can change to this directory at the MATLAB prompt by typing: cd c:\matlab\truss\fem

Step 4. Type: nrlfemi at the MATLAB prompt. You will now see the following menu (in color):



Step 5. If a session was not saved (or a new one is desired), click on the *Geometry and Rod Properties* button. This will evoke the *geomet.m* file. MATLAB will prompt the user by displaying text after a >>.

>>Enter the number of nodes:

Type the total number of nodes (52 in the case of the NPS Space Truss) and *[return]*.

>>Input [Node#,X,Y,Z]

Each node will be assigned a unique, arbitrary node #, and then its specific location. i.e. node #1 is at X=0, Y=0, and Z=0 then type:

[1,0,0,0] [return]

etc.

See Appendix C, NRLFEMI – NPS Space Truss Properties, for node locations for the NPS Space Truss.

>>Tap [return] to scroll through node data.

[return]

>>geo=

1 0 0 0

2 1 0 0

: : : :

>>Change anything? (y/n):

The user has an opportunity to retype any erroneous entries by typing *y*. Other wise the user may type *n*.

n [return]

>>Tap [return] to continue.

[return]

>>Enter fixed node numbers, e.g., [1 2 3 4]:

In the case of the NPS Space Truss, nodes 1, 2, 27, & 28 are fixed.

[1, 2, 27, 28] [return]

>>Tap [return] to continue.

[return]

At this point the screen is cleared and the follow text is displayed:

>>CONNECTIONS BETWEEN NODES

Connections between nodes are to be specified one pair

at a time. For example, if nodes 1 and 5 are to be

connected, the required input is [1 5 eid keff weight].

to help keep track of the number of connections, each connection entry will be accompanied by a number, starting with 1 and continuing up through the number of elements. If the last entry line number does not equal the total number of connecting rods, then something is wrong. After the last connection has been input, input [0 0 0 0 0] for the next line.

Enter [From Node, To Node, Element ID, Stiffness (lb/in), Weight (lb)]

1

An element connected between nodes 3 and 4 would be given an arbitrary element i.d. of 1 (all eids must be unique). Its stiffness is 31790.67 lb/in and it weighs 0.03 lb. The entry would be
[3, 4, 31790.67, 0.03] [return]
etc.
After the last entry, type
[0, 0, 0, 0, 0] [return]

>>The basic geometry has now been entered.
Tap [return] to continue.
[return]

>>Save this data to a file geomet.mat? (y/n):
Type *y* to save or *n* to not save. It is a good idea to move the existing truss property files to a different directory (*c:\matlab\Truss\fem\temp(old)* is currently setup for this purpose) in case that data is desired again, it won't need to be retyped.
n

After typing *n*, the pop-up menu will appear on the screen again. If you had typed *y*, then the program would have received a warning stating that the file *geomet.mat* will be overwritten. Then you will be prompted to continue or not. If

you choose to save the current data to the file *geomet.mat*, the old *geomet.ma*t will be replaced with the new one.

From the pop-up menu, it is advised that you point-and-click on the *View Structure* option. Doing so will display a static three-dimensional view of the structure. This is for making sure that you have entered your connections and node points properly.

Next, from the pop-up menu, point-and-click on the *Define Lumped Weights* option. Doing so will bring up the following:

>>DEFINE WEIGHT PROPERTIES
Note: Weight values are to be entered in (lb).

Concentrated weights at each node are to include
(node ball)+(standoffs)+(nuts)+(bolts)+(pins)+(sleeves).

Enter concentrated mass for node 3 (lb):
Simply type in a number for the lumped weight at node 3 in POUNDS!!
0.30 [return]

>>The basic lumped weight have now been entered.
Tap [return] to continue.
Then you will be asked whether you want to save the data or not.

After you answer, the pop-up menu will appear in the corner again. Select the *Compute Mass and Stiffness* selection. The program will now assemble the elemental mass and stiffness matrices and assemble them into the gross mass and stiffness matrices. A count will appear in the left-hand top corner of the screen, telling the element i.d. for which the code is generating mass and stiffness matrices.

After creating the mass and stiffness matrices, you will be asked whether or not you want to save the data. Answer correspondingly. The pop-up menu will appear again.

Now that you have the mass and stiffness matrices, you can calculate the eigenstructure. Point-and-click on the *Calculate Eigenstructure* selection. If you have a large model, this calculation may take some time depending on you PC platform. After the eigenstructure is calculated, you will be asked which frequencies you wish to view. You could type something like *[1:20]* to see the first twenty natural frequencies. You will be asked if you want to save the data to a file. Answer correspondingly.

After you have calculated the eigenstructure, you can animate the modeshapes. Point-and-click on the *Animate Structure* selection. If will take a little while for MATLAB to get the animation sequence together. NOTE: MATLAB requires that your monitor be set to 256 colors to do this animation.

By selecting the *Calculate FRF* option from the pop-up menu, the code will prompt you with node numbers and directions for the out/in FRF. Also, it will ask you about total sample time and number of samples. This just determines the frequency range to plot and the number of points. NOTE: The method implemented by this code is very inefficient and should be redone using the total modal solution as opposed to inverting. However, the inversion method would be good for determining large numbers of FRFs.

APPENDIX B. NRLFEMI MATLAB CODE

This appendix contains the MATLAB code used to generate the calculated NPS Space Truss natural frequencies. It was originally developed by Robert Craig Waner for NRL on 7 August 1995, and later modified by Brent K. Andberg. The following finite element code script files are included in this order:

ANIMATE.M

```
% ANIMATE  Creates a movie of moving structure.  This
%          m-file is called from NRLFEMI.M.

if fg4==1 & fg1==1,
   chkfigs=get(0,'Children');
   if length(chkfigs)>0,
      set(chkfigs,'Visible','Off');
   end

   clc;
   disp('ANIMATE STRUCTURE')
   disp(' ')
   disp('Note: Creating the animation sequence may be fairly time
consuming.')
   mshpnum=input('Enter mode shape number to display: ');
   disp(' ')
   disp('Tap [return] to continue.');pause;clc;
   mshp=phi(:,mshpnum);
   srtchkfg=sort(chkfigs);
   sizesrt=size(srtchkfg);
   figur=srtchkfg(sizesrt(1,1),1)+1;
   cntr1=[];cntr2=0;
   sizecon=size(connection);
   numel=sizecon(1,1);


   for cntr1=0.1:-0.02:-0.1,
      ms=mshp*cntr1;
      cntr2=cntr2+1;
      nodes2=reducedgeo;
      geo2=geo;
      sznodes2=size(nodes2);
      cntr4=[];
      for cntr4=1:1:sznodes2(1,1),
         nodenm=nodes2(cntr4,1);
         xyz=nodes2(cntr4,2:4);
         indndf=find(nodedof(:,1)==nodenm);
         philoc=[nodedof(indndf,2:4)];
       partphi=ms(philoc,1);
         nodes2(cntr4,2:4)=reducedgeo(cntr4,2:4)+partphi';
      end
      cntr4=[];
      for cntr4=1:1:sznodes2(1,1),
         nodenm2=nodes2(cntr4,1);
         rplace=nodes2(cntr4,2:4);
         ind2geo=find(geo2(:,1)==nodenm2);
         geo2(ind2geo,2:4)=rplace;
      end
      cntr4=[];
      for cntr4=0:1:(numel-1),
         fromnode=connection((cntr4+1),1);
         tonode=connection((cntr4+1),2);
         fromnodeind=find(geo2(:,1)==fromnode);
         tonodeind=find(geo2(:,1)==tonode);
```

```
lines((cntr4*2+1):(cntr4*2+2),:)=[geo2(fromnodeind,2:4);geo2(tonodeind,2
:4)];
      end
      cntr4=[];
      figure(figur);clf;
      for cntr4=0:1:(numel-1),
plot3(lines((cntr4*2+1):(cntr4*2+2),1),lines((cntr4*2+1):(cntr4*2+2),2),
lines((cntr4*2+1):(cntr4*2+2),3));
       set(figur,'Position',[3 118 343 301],'Visible','Off');
       if cntr4==0,
          xlabel('X');ylabel('Y');zlabel('Z');axis([mincoord maxcoord
mincoord maxcoord mincoord maxcoord]);
          hold on;
       end
       end
       set(figur,'Visible','On','Color',[0 0 0]);
      home; disp(sprintf('Frame #: %g',cntr2'))
      cntr3=[];
      for cntr3=0:1:(numel-1),
      end
      Movmat1(:,cntr2)=getframe;
   end

% Arrange Movmat for smooth playback.
   Movmat2=fliplr(Movmat1);
   szm=size(Movmat2);
   Movmat2(:,[1 szm(1,2)])=[];
   Movmat=[Movmat1,Movmat2];
   clear Movmat1 Movmat2

% Play movie.
   movie(Movmat,20);
   callback=['set(uip,''back'',[1 2/3 1/3]);' ...
       'movie(Movmat,20);' ...
       'set(uip,''back'',''default'');'];
   uip=uicontrol('style','push','pos',[20 10 60
40],'string','Replay','call',callback);
   uiq=uicontrol('style','push','pos',[100 10 60
40],'string','Done','call','clf;');
else,

% Inform user insufficient data for animation.
   chkfigs=get(0,'Children');
   if length(chkfigs)>0,
      set(chkfigs,'Visible','Off');
   end
   clc;
   if fg4~=1,
      disp('Eigenstructure has not been computed')
      disp('or loaded.  The eigenstructure can be')
      disp('computed by choosing the Calculate')
      disp('Eigenstructure option.  A previously')
      disp('calculated eigenstrucure may be loaded')
      disp('by choosing the Load Previous Session')
      disp('option.')
   end
```

```
   if fg1~=1,
      disp('Geometry has not been defined.')
      disp('Either load a previous session, or')
      disp('choose the Geometry And Rod Properties')
      disp('option.')
   end
end
```

CALCEIG.M

```
function [lambda,phi,psi]=calceig(kg,mg)

% CALCEIG Computes the eigenvalue matrix, left & right
%       modal matrices.  This function is called from
%       CALEIG0.M.

% Initialize matrices.
   n=max(size(kg));
   lambda=zeros(n,1);
   phi=zeros(n);
   psi=phi;
   [vleft,dleft]=eig(kg',mg');
   [vright,dright]=eig(kg,mg);
   kr=zeros(n,1);
   kc=kr;
   er=kr;
   ec=kr;

% Sort eigenvectors and eigenvalues.
   indr=0;
   indc=0;
   for i=1:n,
      if abs(imag(dright(i,i)))<=1.e-7,
       indr=indr+1;
       kr(indr)=i;
       er(indr)=dright(i,i);
      elseif imag(dright(i,i))>1.e-7,
       indc=indc+1;
       kc(indc)=i;
       ec(indc)=dright(i,i);
      end
   end
   er=real(er(1:indr));
   ec=ec(1:indc);
   ind=1;
   [lr,krn]=sort(er);
   [lc,kcn]=sort(imag(ec));
   for i=1:(indr+indc),
      if i<=indr,
       phi(:,i)=real(vright(:,kr(krn(i))));
       lambda(i)=real(dright(kr(krn(i)),kr(krn(i))));
       ind=ind+1;
      else,
       ii=i-indr;
       phi(:,ind)=vright(:,kc(kcn(ii)));
       phi(:,ind+1)=conj(phi(:,ind));
       lambda(ind)=dright(kc(kcn(ii)),kc(kcn(ii)));
       lambda(ind+1)=conj(lambda(ind));
       ind=ind+2;
      end
   end

   indr=0;
   indc=0;
   for i=1:n;
```

```
      if abs(imag(dleft(i,i)))<=1e-7,
       indr=indr+1;
       kr(indr)=i;
       er(indr)=dleft(i,i);
      elseif imag(dleft(i,i))>1e-7,
       indc=indc+1;
       kc(indc)=i;
       ec(indc)=dleft(i,i);
      end
   end
   er=real(er(1:indr));
   ec=ec(1:indc);
   ind=1;
   [lr,krn]=sort(er);
   [lc,kcn]=sort(imag(ec));
   for i=1:(indr+indc),
      if i<=indr,
       psi(:,i)=real(vleft(:,kr(krn(i))));
       ind=ind+1;
      else
       ii=i-indr;
       psi(:,ind)=vleft(:,kc(kcn(ii)));
       psi(:,ind+1)=conj(psi(:,ind));
       ind=ind+2;
      end
   end

% Normalization for unit modal mass.
   for i=1:n,
      xi=phi(:,i);
      yi=psi(:,i);
      sc1=conj(xi')*mg*xi;
      phi(:,i)=phi(:,i)/(sqrt(sc1));
      sc2=conj(yi')*mg*phi(:,i);
      psi(:,i)=psi(:,i)/sc2;
   end
```

CALEIGO.M

```
% CALEIG0 Compute eigenstructure.  This m-file is called
%       from NRLFEMI.M.

if fg3==1,
   chkfigs=get(0,'Children');
   if length(chkfigs)>0,
      set(chkfigs,'Visible','Off');
   end
   clc;
   disp('COMPUTE EIGENSTRUCTURE')
   [lambda,phi,psi]=calceig(kgross,mgross);
   wn=lambda .^0.5;
   freqhz=wn/(2*pi);
   szfr=size(freqhz);
   disp(' ');
   nts=input(sprintf('Enter range of undamped natural freq to list (max
%g): ',szfr(1,1)));
   snts=size(nts);
   indxr=[nts(1,1):1:nts(1,snts(1,2))]';
nflist=[indxr,wn(nts(1,1):nts(1,snts(1,2)),1),freqhz(nts(1,1):nts(1,snts
(1,2)),1)];
   format bank;
   disp(' ');
   disp('         NUMBER     wn(rad/s)      freq(Hz)')
   disp(nflist)
   format;
   disp('Tap [return] to continue.');pause;clc;
   yn=[];
   yn=input('Save this data to a file estruct.mat? (y/n): ','s');
   if yn=='y',
      disp(' ');
      disp('Warning: File estruct.mat will be overwritten!');
      disp(' ');
      yna=input('Continue? (y/n): ','s');
      if yna=='y',
         chdir c:\matlab\truss\fem\temp
         save estruct lambda psi phi freqhz wn
         chdir c:\matlab\truss\fem
      end
   end
   fg4=1;

   set(2,'Visible','on')

else,
% Inform user that kgross and mgross are needed to
% calculate the eigenstructure.
   chkfigs=get(0,'Children');
   if length(chkfigs)>0,
       set(chkfigs,'Visible','Off');
   end
   clc;
   disp('The gross mass and stiffness')
   disp('matrices must be assembled or')
   disp('loaded.  The gross mass and')
```

```
   disp('stiffness may be assembled by')
   disp('choosing Compute Mass and Stiffness.')
   disp('Previously assembled gross mass and')
   disp('stiffness may be loaded under the')
   disp('Load Previous Session option.')
end
```

DEFPROP.M

```
% DEFPROP Define lumped weights for Naval Research
%       Lab truss-structure.  This m-file is called
%       from NRLFEMI.M.  The weights of the nodeballs,
%       standoffs, nuts, sleeves, and epoxy will be
%       lumped as point masses.
%
% Note:  Weight values must be entered in (lb).

clc;
chkfigs=get(0,'Children');
if length(chkfigs)>0,
   set(chkfigs,'Visible','Off');
end
sizecon=size(connection);
cntr=[]; eidvect=[]; nodedof=[]; indrmv=[]; connotinc=[];
reducedconn=[]; dofs1=[]; dof2=[];
sizegeo=size(geo);
sizefixednodes=size(fixednodes);
reducedsize=sizegeo(1,1)-sizefixednodes(1,2);
reducedgeo=geo;
for cntr=1:1:sizefixednodes(1,2),
   indrmv(cntr,1)=find(geo(:,1)==fixednodes(1,cntr));
end
reducedgeo(indrmv,:)=[];
cntr=[];
for cntr=1:1:reducedsize,
   nodedof(cntr,:)=[reducedgeo(cntr,1),(cntr*3-2),(cntr*3-1),(cntr*3)];
end
cwghtmat=[];
disp('DEFINE WEIGHT PROPERTIES')
disp('Note:  Weight values are to be entered in (lb).')
disp(' ')
disp('Concentrated weights at each node are to include')
disp('(node ball)+(standoffs)+(nuts)+(bolts)+(pins)+(sleeves).')
sizenodedof=size(nodedof);
cntr=[];
for cntr=1:1:sizenodedof(1,1),
   conweight=input(sprintf('Enter concentrated mass for node %g (lb):
',nodedof(cntr,1)));
   cwghtmat(cntr,1:2)=[nodedof(cntr,1),conweight];
end

disp(' ')
disp('The basic lumped weights have now been entered.')
disp('Tap [return] to continue.');pause;
clc;
yn=[];
yn=input('Save the lumped weight data to a file propert.mat? (y/n):
','s');
if yn=='y',
   disp(' ');
   disp('Warning: File propert.mat will be overwritten!');
   disp(' ');
   yna=input('Continue? (y/n): ','s');
   if yna=='y',
```

```
      chdir c:\matlab\truss\fem\temp
      save propert cwghtmat
      chdir c:\matlab\truss\fem
   end
end
fg2=1;  %flag
```

FRESP.M

```
% FRESP Calculate frequency response function.
%     This m-file is called from NRLFEMI.M.

   chkfigs=get(0,'Children');
   if length(chkfigs)>0,
      set(chkfigs,'Visible','Off');
   end
   clc;
   disp('CALCULATE ANALYTICAL FREQUENCY RESPONSE FUNCTION')
   disp(' ')

%%%temporarily
   dgross=0.0000001*kgross;

   numsamp=input('Enter number of samples: ');
   ttime=input('Enter total sample time (sec): ');
   nl=numsamp+1;
   fmax=numsamp/ttime;
   df=1/ttime;
   f=(0:numsamp)*df;
   disp(' ')
   disp(sprintf('The FRF will be computed from 0 to %g (Hz).',fmax))
   disp(sprintf('Points on the FRF will be computed every  %g Hz).',df))
   disp(' ')
   disp('Tap [return] to continue.'); pause;clc;
   cntr1=[];cntr2=[];
   cntr2n=input('Enter node number for excitation (input): ');
   direct2=input('Enter direction of excitation (x,y,or z): ','s');
   indinput=find(nodedof(:,1)==cntr2n);
   if direct2=='x',
      cntr2=nodedof(indinput,2);
   elseif direct2=='y',
      cntr2=nodedof(indinput,3);
   else,
      cntr2=nodedof(indinput,4);
   end
   cntr1n=input('Enter node number for response (output): ');
   direct1=input('Enter direction of response (x,y, or z): ','s');
   indoutput=find(nodedof(:,1)==cntr1n);
   if direct1=='x',
      cntr1=nodedof(indoutput,2);
   elseif direct1=='y',
      cntr1=nodedof(indoutput,3);
   else,
      cntr1=nodedof(indoutput,4);
   end
   disp(' ')
   disp(sprintf('Transfer function to be calculated is
(%g%s/%g%s).',cntr1n,direct1,cntr2n,direct2))
   disp(' ')
   disp('Tap [return] to start calculation of FRF.'); pause;
   for n=1:nl,
      H(n)=xfer(cntr1,cntr2,(n-1)*df,mgross,dgross,kgross);
   end
```

```
   srtchkfg=sort(chkfigs);
   sizesrt=size(srtchkfg);
   fig=srtchkfg(sizesrt(1,1),1)+1;
   labelt=sprintf('FRF for Rsp/Exc:
(%g%s/%g%s)',cntr1n,direct1,cntr2n,direct2)
   figure(fig);clf;subplot(211);
   semilogy(f(2:nl),abs(H(2:nl)));title(labelt);xlabel('Frequency
(Hz)');
   ylabel('Inertance (in/s^2/lb)'); grid;
   subplot(212);
   anglefrf=angle(H);
   angfrfd=anglefrf*180/pi;
   plot(angfrfd,f);
   plot(f,angfrfd);axis([0 max(f) -10 200]);grid;
   set(2,'Visible','On');
```

GEOMET.M

```
% GEOMET Set up geometry for Naval Reseach Lab truss structure.
%       GEOMET prompts for the number of nodes and their
%       locations in a global sense.  For consistency, global
%       locations should be in (inches).  This m-file is
%       called from NRLFEMI.M.

% INPUT NODE LOCATIONS.
   geo=[]; connection=[];
   set(gcf,'Visible','off');
   clc; nn=[];
   nn=input('Enter the number of nodes: ');
   cntr=0;
   disp(' ')
   disp('Input [Node#,X,Y,Z]')
   for cntr=1:1:nn,
      geoi=input('');
      sizegeoi=size(geoi);
      if sizegeoi(1,2)==4,
         geo(cntr,:)=geoi;
      end
      while sizegeoi(1,2)~=4,
        disp('Improper input! Re-enter data.')
        geoi=input('');
        sizegeoi=size(geoi);
        if sizegeoi(1,2)==4,
           geo(cntr,:)=geoi;
        end
      sizegeo=size(geo);
      end
   end
   disp(' ')
   disp('Tap [return] to scroll through node data.');
   pause;clc;
   cntr=0;
   if nn>25,
      nnd=fix(nn/25);
      for cntr=0:1:(nnd-1),
         disp(geo((cntr*25+1):(cntr*25+25),:))
         disp('Tap [return] to continue scrolling...'); pause;
      end
      disp(geo((cntr*25+26):nn,:))
   else,
      geo
   end
   yn='y';
   while yn=='y',
      yn=input('Change anything? (y/n): ','s');
      if yn=='y',
         nntc=input('Enter node number to change: ');
         ind=find(geo(:,1)==nntc);
         if ind==[],
            disp('This node number is not contained in the data!')
         else,
            nxyz=input('Enter new [X,Y,Z] coordinates: ');
            geo(ind,1:4)=[nntc,nxyz];
```

```
        end
      end
    end
    disp(' ')
    disp('Tap [return] to continue.'); pause;
    clc;
    fixednodes=input('Enter fixed node numbers, e.g., [1 2 3 4]: ');
    disp(' ')
    disp('Tap [return] to continue.'); pause;clc;

% INPUT NODAL CONNECTIONS.
    connections=[];
    disp('CONNECTIONS BETWEEN NODES');
    disp(' ')
    disp('Connections between nodes are to be specified one pair')
    disp('at a time.  For example, if nodes 1 and 5 are to be')
    disp('connected, the required input is [1 5 eid keff weight].')
    disp('to help keep track of the number of connections, each
connection')
    disp('entry will be accompanied by a number, starting with 1')
    disp('and continuing up through the number of elements.  If')
    disp('the last entry line number does not equal the total')
    disp('number of connecting rods, then something is wrong.')
    disp('After the last connection has been input, input')
    disp('[0 0 0 0 0] for the next line.')

    connection=[1 1 1 1 1]; cntr=0;
    disp(' ')
    disp('Enter [From Node, To Node, Element ID, Stiffness (lb/in),
Weight (lb)]')
    sizecon=size(connection);
    while connection(sizecon(1,1),:)~=[0 0 0 0 0],
       cntr=cntr+1;
       connection(cntr,:)=input(sprintf('%g  ',cntr));
       sizecon=size(connection);
    end
    sizefix=size(fixednodes);
    ndof=(nn-sizefix(1,2))*3;
    connection(sizecon(1,1),:)=[];
    sizecon=size(connection);
    numel=sizecon(1,1);
    cntr=[];
    for cntr=0:1:(numel-1),
       fromnode=connection((cntr+1),1);
       tonode=connection((cntr+1),2);
       fromnodeind=find(geo(:,1)==fromnode);
       tonodeind=find(geo(:,1)==tonode);
lines((cntr*2+1):(cntr*2+2),:)=[geo(fromnodeind,2:4);geo(tonodeind,2:4)]
;
    end
    mincoord=min(min(lines));
    maxcoord=max(max(lines));
    disp(' ')
    disp('The basic geometry has now been entered.')
    disp('Tap [return] to continue.');pause;
    clc;
```

```
yn=[];
yn=input('Save this data to a file geomet.mat? (y/n): ','s');
if yn=='y',
   disp(' ');
   disp('Warning: File geomet.mat will be overwritten!');
   disp(' ');
   yna=input('Continue? (y/n): ','s');
   if yna=='y',
      chdir c:\matlab\truss\fem\temp
      save geomet geo connection fixednodes ndof mincoord maxcoord
      chdir c:\matlab\truss\fem
   end
end
fg1=1;

set(2,'Visible','on')
```

MANDK.M

```
% MANDK Compute gross mass and stiffness matrices for
%      Naval Research Lab truss-structure.  This m-file
%      is called from NRLFEM.M and requires that the
%      geometry, stiffness, weights, and lumped weights
%      be defined prior to execution.  For an overview
%      of this FEM code, see the notes in nrlfemi.m.

   if fg1==1 & fg2==1,  %flags to check if geometry & props available
      chkfigs=get(0,'Children');
      if length(chkfigs)>0,
         set(chkfigs,'Visible','Off');
      end
      sizecon=size(connection);
      cntr=[]; eidvect=[]; nodedof=[]; indrmv=[]; connotinc=[];
      reducedconn=[]; dofs1=[]; dof2=[];kssparse=[];
      sizegeo=size(geo);
      sizefixednodes=size(fixednodes);
      reducedsize=sizegeo(1,1)-sizefixednodes(1,2);
      reducedgeo=geo;
      for cntr=1:1:sizefixednodes(1,2),
       indrmv(cntr,1)=find(geo(:,1)==fixednodes(1,cntr));
      end
      reducedgeo(indrmv,:)=[];
      cntr=[];

%    Assign global degrees of freedom to free nodes.
      for cntr=1:1:reducedsize,
         nodedof(cntr,:)=[reducedgeo(cntr,1),(cntr*3-2),(cntr*3-
1),(cntr*3)];
      end

%  Determine which rod elements not to include in Mgross and Kgross.
    jj=fixednodes';
    cntr=[];
    mat1=[];
    mat2=[];
    sizejj=size(jj);
    for cntr=sizejj(1,1):-1:1,
      mat2=[mat2;mat1];
      mat1=[jj((sizejj(1,1)-cntr+1),1)*ones((cntr-1),1),jj((sizejj(1,1)-
cntr+2):sizejj(1,1) ,1)];
    end
    connotinc=[mat2;[mat2(:,2),mat2(:,1)]];
    sizeconnotinc=size(connotinc);
    cntr=[];cntr2=1;cntr3=[];
    for cntr=1:1:sizeconnotinc(1,1),
       connotincchk=connotinc(cntr,:);
       for cntr3=1:1:sizecon(1,1),
          if connection(cntr3,1:2)==connotincchk,
           indnot(cntr2,1)=cntr3;
                 cntr2=cntr2+1;
          end
       end
    end
    reducedconn=connection;
```

```
    reducedconn(indnot,:)=[];
    sizeredcon=size(reducedconn);
    cntr=[];

%  Determine which arguments to pass to the function strutkw.
       for cntr=1:1:sizeredcon(1,1),
        fromnode=reducedconn(cntr,1);
        tonode=reducedconn(cntr,2);
        indfromnode=find(geo(:,1)==fromnode);
        indtonode=find(geo(:,1)==tonode);
        vctr=geo(indtonode,2:4)-geo(indfromnode,2:4);
        normvctr=vctr/(norm(vctr));
        eidvect(cntr,:)=[reducedconn(cntr,3),normvctr];
       end
       cntr=[];
       sizeeidvect=size(eidvect);
%  Actually construct gross stiffness and mass matrices for rods.
%  Note: Lumped masses to be added later.
       mrods=zeros(ndof,ndof);
       krods=zeros(ndof,ndof);
       for cntr=1:1:sizeeidvect(1,1),
        dofs1=[]; dofs2=[]; ms=[]; ks=[];
        home; eid=eidvect(cntr,1)
        indprop=find(reducedconn(:,3)==eid);
          node1=reducedconn(indprop,1);
        node2=reducedconn(indprop,2);
        indnode1=find(nodedof(:,1)==node1);
        dofs1=nodedof(indnode1,2:4);
          indnode2=find(nodedof(:,1)==node2);
        dofs2=nodedof(indnode2,2:4);
        stiff=reducedconn(indprop,4);
        wght=reducedconn(indprop,5);
%  Determine which transformation matrix to apply.
        ost=1/(sqrt(2)); dcm=[];
        if abs(eidvect(cntr,2:4))==[1 0 0],
             dcm=[1 0 0;0 0 -1;0 1 0];
        elseif abs(eidvect(cntr,2:4))==[0 1 0],
             dcm=[0 1 0;1 0 0;0 0 -1];
        elseif abs(eidvect(cntr,2:4))==[0 0 1],
           dcm=[0 0 1;1 0 0;0 1 0];
          elseif eidvect(cntr,2:4)==[ost ost 0],
           dcm=[ost ost 0;ost (-ost) 0;0 0 (-1)];
        elseif eidvect(cntr,2:4)==(-1)*[ost ost 0],
           dcm=[ost ost 0;ost (-ost) 0;0 0 (-1)];
          elseif eidvect(cntr,2:4)==[ost (-ost) 0],
           dcm=[ost (-ost) 0;ost ost 0;0 0 1];
        elseif eidvect(cntr,2:4)==(-1)*[ost (-ost) 0],
           dcm=[ost (-ost) 0;ost ost 0;0 0 1];
        elseif eidvect(cntr,2:4)==[0 ost ost],
           dcm=[0 ost ost;1 0 0;0 ost (-ost)];
        elseif eidvect(cntr,2:4)==(-1)*[0 ost ost],
           dcm=[0 ost ost;1 0 0;0 ost (-ost)];
        elseif eidvect(cntr,2:4)==[0 (-ost) ost],
           dcm=[0 (-ost) ost;1 0 0;0 ost ost];
        elseif eidvect(cntr,2:4)==(-1)*[0 (-ost) ost],
           dcm=[0 (-ost) ost;1 0 0;0 ost ost];
        elseif eidvect(cntr,2:4)==[ost 0 ost],
```

```
          dcm=[ost 0 ost;ost 0 (-ost);0 1 0];
       elseif eidvect(cntr,2:4)==(-1)*[ost 0 ost],
          dcm=[ost 0 ost;ost 0 (-ost);0 1 0];
       elseif eidvect(cntr,2:4)==[(-ost) 0 ost],
          dcm=[ost 0 (-ost);ost 0 ost;0 (-1) 0];
       elseif eidvect(cntr,2:4)==(-1)*[(-ost) 0 ost],
          dcm=[ost 0 (-ost);ost 0 ost;0 (-1) 0];
       else
          disp(' ')
          error('This program was written assuming square bays.')
       end
       if isempty(dofs1) | isempty(dofs2),
          dummydof=[(ndof+1):(ndof+3)];
          if isempty(dofs1),
             lev=[dofs2,dummydof];
             [msub,ksub]=strutkw(dcm,lev',ndof,stiff,wght);
             msub=msub(1:ndof,1:ndof);
             ksub=ksub(1:ndof,1:ndof);
          end
          if isempty(dofs2),
             lev=[dofs1,dummydof];
             [msub,ksub]=strutkw(dcm,lev',ndof,stiff,wght);
             msub=msub(1:ndof,1:ndof);
             ksub=ksub(1:ndof,1:ndof);
            end
         else,
            lev=[dofs1,dofs2];
          [msub,ksub]=strutkw(dcm,lev',ndof,stiff,wght);
         end
%%%
         kssparse=sparse([kssparse;ksub]);
         mrods=mrods+msub;
         krods=krods+ksub;
      end
         kgross=krods;
         clear krods;

%  Create concentrated mass matrix.
      sizecwghtmat=size(cwghtmat);
      cntr=[];
      for cntr=1:1:sizecwghtmat(1,1),
       nodeind=find(nodedof(:,1)==cwghtmat(cntr,1));
       dofp=nodedof(nodeind,2:4);
       weight=cwghtmat(cntr,2);
       conv2mass=weight/386.4;
       conm(dofp,dofp)=conv2mass*eye(3);
      end

      mgross=mrods+conm;

%  Save mgross, kgross, nodedof, reducedgeo to file.
   disp(' ')
   disp('The gross mass and stiffness matrices have been assembled.')
   disp('Tap [return] to continue.');pause;
   clc;

   yn=[];
```

```
   yn=input('Save this data to a file mkgross.mat? (y/n): ','s');
   if yn=='y',
      disp(' ');
      disp('Warning: File mkgross.mat will be overwritten!');
      disp(' ');
      yna=input('Continue? (y/n): ','s');
      if yna=='y',
         chdir c:\matlab\truss\fem\temp
         save mkgross mgross kgross nodedof reducedgeo reducedconn
kssparse
         chdir c:\matlab\truss\fem
      end
   end
   fg3=1;

   set(2,'Visible','on')

%  Inform user that there is insufficient data to construct
%  mass and stiffness matrices.
   else,
    chkfigs=get(0,'Children');
    if length(chkfigs)>0,
       set(chkfigs,'Visible','Off');
    end
    clc;
    disp('Geometry and rod properties and/or')
    disp('lumped weights have not been defined!')
    disp('Either load previous sessions, or')
    disp('define geometry and rod properties')
    disp('and/or lumped weights.')
   end
```

NRLFEMI.M

```
% NRLFEMI  Finite element model for Naval Research Lab truss structure.
%         NRLFEMI requires spatial and material information from a data
%         deck and builds the gross stiffness and mass matrices.  Units
%         of stiffness are to be (lb/in) and units of weight are to be
%         in (lb).
%
%           This program calls on the following m-files:
%           animate.m Animates the structure for specified mode shape.
%           calceig.m Function that calculates eigenstructure.
%           caleig0.m Calls calceig.m & displays range of nat. freqs.
%           comment.m Function that tells user that a file is loaded.
%           defprop.m User enters node numbers and lumped masses.
%           fresp.m   Calculates & displays user-chosen frf. Calls
xfer.m.
%           geomet.m  Prompts user to input node locations and
connections.
%           mandk.m   Calls strutkw.m & calculates gross mass & stiff
mat.
%           prevses.m User chooses previous session parameters to load.
%           strutkw.m Function that returns elemental m & k to mandk.m.
%           viewstr.m Displays static structure configurtaion.
%           xfer.m    Function that returns H(w) to fresp.m.
%
%         All of the above m-files and this one (nrlfemi.m) should
%         be located in a directory called matlab\truss\fem.  This
%         requirement has to do with loading previous sessions.  If it is
%         desired not to have the path to these m-files be
%         c:\matlab\truss\fem, then appropriate modifications must be
made
%         to the prevses.m file.
%
%         All saved files are directed along the path
%         c:\matlab\truss\fem\temp.  So, there must also be a directory
%         entitled matlab\truss\fem\temp.  If this pathway is not
desired,
%         then appropriate modifications must be made to the prevses.m
file,
%         and all files that prompt the user for saving data.
%
%         This set of m-files assumes that the space truss is composed
%         of cubic bays.  Any deviation from this requirement will
%         result in an error during execution of the m-file, mandk.m.
%         To circumvent the cubic bay assumption, the appropriate
%         modifications to the mandk.m file must be made.  To modify
%         mandk.m, different transformation matrices must be
%         defined.  The generation of the transformation matrices
%         could be easily generalized, but this was not done to save
%         time in the coding of this FEM.  It should be simple to
%         define the transformation matrices using appropriate
%         transformation matrices.
%
%         The elemental mass matrices generated by this finite element
%         code are constructed using a local coupled mass matrix, as
%         opposed to a local lumped mass matrix.  If another local
%         mass matrix is desired, then the appropriate modifications
```

```
%        must be made to strutkw.m.
%
%        NRLFEMI was written for a screen resolution of 640x480.
%        If this program is run on any video but 640x480, the
%        graphical user interfaces defined in this program
%        will not be properly sized and will be unusable.  If
%        this resolution constraint is a problem, commenting
%        out or modifying lines with the 'Position' property
%        in them will help--though figures will most likely
%        run into each other and will have to be moved by
%        dragging them.

% Written by: Robert Craig Waner
% Modified by: Brent K. Andberg

% Date last modified: 10 July 1997

% Menu
   clc; fg1=0; fg2=0; fg3=0; fg4=0;
   header='NRL Finite Element Model';
   labels=str2mat('Load Previous Session','Geometry And Rod
Properties','View Structure','Define Lumped Weights', ...
                'Compute Mass and Stiffness','Calculate
Eigenstructure','Animate Structure','Calculate FRF');
   callbacks=str2mat('prevses','geomet','viewstr', ...
                    'defprop','mandk','caleig0','animate','fresp');
   choices('NRLFEM', header, labels, callbacks);
   set(2,'position',[354 167 264 252]);
```

PREVSES.M

```
% PREVSES Loads a previous session for NRLFEMI.M

openfig=get(0,'Children');
openfigs=sort(openfig);
if length(openfigs)==2,
   close(1);
end
figure(3);clf;

previous=uicontrol(gcf,'Style','text','Position',[0 .8 1
.2],'Units','normalized','String','Select Filetypes to Load');
set(gcf,'MenuBar','None','NumberTitle','Off','Position',[354 40 264
99],'Color',[.5 .5 .5])
cb_geom=uicontrol(gcf,'Style','checkbox','Position',[0.05 0.65 1
.2],'Units','normalized','String','Geometry,Stiffness,Weights','CallBack
','geo=[];connection=[];chdir c:\matlab\truss\fem\temp;load
geomet.mat;chdir c:\matlab\truss\fem;fg1=1;comment(0);');
cb_prop=uicontrol(gcf,'Style','checkbox','Position',[0.05 0.45 1
.2],'Units','normalized','String','Lumped
Weights','CallBack','cwghtmat=[];chdir c:\matlab\truss\fem\temp;load
propert.mat;chdir c:\matlab\truss\fem;fg2=1;comment(1);');
cb_mk=uicontrol(gcf,'Style','checkbox','Position',[0.05 0.25 1
.2],'Units','normalized','String','Mgross and
Kgross','CallBack','mgross=[];kgross=[];chdir
c:\matlab\truss\fem\temp;load mkgross.mat;chdir
c:\matlab\truss\fem;fg3=1;comment(2);');
cb_eign=uicontrol(gcf,'Style','checkbox','Position',[0.05 0.05 1
.2],'Units','normalized','String','Eigenstructure','CallBack','estrut=[]
;chdir c:\matlab\truss\fem\temp;load estruct.mat;chdir
c:\matlab\truss\fem;fg4=1;comment(3)');
```

STRUTKW.M

```
function [mel,kel]=strutkw(dcm,lev,ndof,stiff,wght);

% STRUTKW Create elemental stiffness and mass
%         matrices for Naval Research Lab truss-structure.
%         Elemental mass matrices are created using a
%         coupled mass matrix.  This function is called
%         from MANDK.M.
%
%         Note:  Stiffness is in (lb/in).
%                Weight is in (lb).

% Definitions:
%   dcm    3 by 3 direction cosine matrix
%   lev    6 by 1 locator vector
%   ndof   number of global degrees of freedom
%   stiff  effective stiffness of rod element (lb/in)
%   wght   weight of rod element (lb)

% Local stiffness matrix.
    pst=[1 0 0];
    k=stiff*[diag(pst),diag((-1)*pst);diag((-1)*pst),diag(pst)];

% Local coupled mass matrix.
%     dgm=[5 6 6];
%     m=(wght/(386.4*12))*[diag(dgm),diag(pst);diag(pst),diag(dgm)];

% Local lumped mass matrix.  By commenting the previous two
% lines of script and uncommenting the following, the local
% lumped mass matrix will be used in the FEM formulation
% instead of the local coupled mass matrix.
   dgm=[6 6 6];
   m=(wght/(386.4*12))*[diag(dgm),diag([0 0 0]);diag([0 0
0]),diag(dgm)];

% Transformation from local to global coordinates.
    t=[dcm,zeros(3);zeros(3),dcm];
    kt=t'*k*t;
    mt=t'*m*t;

% Expand to system DOF's.
    le=zeros(6,ndof);
    cntr=[];
    for cntr=1:6,
       le(cntr,lev(cntr))=1;
    end
    kel=le'*kt*le;
    mel=le'*mt*le;
```

VIEWSTR.M

```
% VIEWSTR View static structure.
%       VIEWSTR brings up a 3-D figure, showing the
%       static truss structure.  This m-file is
%       called from NRLFEMI.M.

   if fg1==1,
    sizecon=size(connection);
    numel=sizecon(1,1);
    cntr=[];
    for cntr=0:1:(numel-1),
       fromnode=connection((cntr+1),1);
       tonode=connection((cntr+1),2);
       fromnodeind=find(geo(:,1)==fromnode);
       tonodeind=find(geo(:,1)==tonode);
lines((cntr*2+1):(cntr*2+2),:)=[geo(fromnodeind,2:4);geo(tonodeind,2:4)]
;
    end
    mincoord=min(min(lines));
    maxcoord=max(max(lines));
    figure(1);clf;
    set(1,'Visible','Off');
    cntr=[];
    for cntr=0:1:(numel-1),
plot3(lines((cntr*2+1):(cntr*2+2),1),lines((cntr*2+1):(cntr*2+2),2),line
s((cntr*2+1):(cntr*2+2),3));
       set(gcf,'Position',[3 118 343 301]);
       if cntr==0,
       axis([mincoord maxcoord mincoord maxcoord mincoord maxcoord]);
       xlabel('X');ylabel('Y');zlabel('Z');
       hold on;
       end
    end
    set(1,'Visible','On','Color',[0 0 0]);
   else,
    chkfigs=get(0,'Children');
    if length(chkfigs)>0,
       set(chkfigs,'Visible','Off');
    end
    clc;
    disp('Geometry has not been defined!')
    disp('Either load a previous session, or')
    disp('define geometry, stiffness & weights.')
   end
```

XFER.M

```
function H=xfer(m,n,f,M,C,K)

% XFER Computes the frequency response function.  This
%      function is called from FRESP.M.

  s=j*2*pi*f;
  T=inv(s*s*M+s*C+K);
  H=s*s*T(m,n);
```

APPENDIX C. NRLFEMI – NPS SPACE TRUSS PROPERTIES

This appendix contains data from the MATLAB *.mat* files, necessary to build a model of the NPS Space Truss.

The following are the geometric values entered in the NRLFEMI program to describe the location of the node balls on NPS Space Truss (saved in *geomet.mat*):

	Location		
Node #	**X**	**Y**	**Z**
1	0	0	0
2	1	0	0
3	-5	1	0
4	-4	1	0
5	-3	1	0
6	-2	1	0
7	-1	1	0
8	0	1	0
9	1	1	0
10	2	1	0
11	3	1	0
12	4	1	0
13	5	1	0
14	6	1	0
15	-5	2	0
16	-4	2	0
17	-3	2	0
18	-2	2	0
19	-1	2	0
20	0	2	0
21	1	2	0
22	2	2	0
23	3	2	0
24	4	2	0
25	5	2	0
26	6	2	0
27	0	0	-1
28	1	0	-1
29	-5	1	-1
30	-4	1	-1
31	-3	1	-1
32	-2	1	-1
33	-1	1	-1
34	0	1	-1
35	1	1	-1
36	2	1	-1
37	3	1	-1

38	4	1	-1
39	5	1	-1
40	6	1	-1
41	-5	2	-1
42	-4	2	-1
43	-3	2	-1
44	-2	2	-1
45	-1	2	-1
46	0	2	-1
47	1	2	-1
48	2	2	-1
49	3	2	-1
50	4	2	-1
51	5	2	-1
52	6	2	-1

The following are the individual node, combined masses (saved in *propert.mat*). These masses include the accelerometer mass (or dummy mass), any extra thumb screws (impact points), and all associated end assemblies.

Note: In the NRLFEMI program, masses are called for in units of pounds mass.

Node #	mass [lbs.]
3.0000	0.3136
4.0000	0.5429
5.0000	0.3518
6.0000	0.5429
7.0000	0.3518
8.0000	0.6576
9.0000	0.3901
10.0000	0.5429
11.0000	0.3518
12.0000	0.5429
13.0000	0.3518
14.0000	0.4283
15.0000	0.4283
16.0000	0.3518
17.0000	0.5429
18.0000	0.3518
19.0000	0.5429
20.0000	0.3518
21.0000	0.5429
22.0000	0.3518
23.0000	0.5429
24.0000	0.3801
25.0000	0.5429
26.0000	0.3136
29.0000	0.4283
30.0000	0.3518
31.0000	0.5429
32.0000	0.3518

33.0000	0.5429
34.0000	0.3901
35.0000	0.6576
36.0000	0.3518
37.0000	0.5429
38.0000	0.3518
39.0000	0.5429
40.0000	0.3136
41.0000	0.3418
42.0000	0.5429
43.0000	0.3518
44.0000	0.5429
45.0000	0.3518
46.0000	0.5429
47.0000	0.3518
48.0000	0.5429
49.0000	0.3518
50.0000	0.5429
51.0000	0.3518
52.0000	0.4283

The following is a description of the connection between nodes as required by NRLFEMI to build the NPS Space Truss FEM (the elements have been numbered arbitrarily):

				[lb/in]	[lb]
Fm node	To node	Element #	Type	stiffness	MASS
1	2	1	Longeron	2.95E+04	0.030534
1	8	2	Longeron	2.95E+04	0.030534
1	27	3	Longeron	2.95E+04	0.030534
2	8	4	Diagonal	1.94E+04	0.046848
2	9	5	Longeron	2.95E+04	0.030534
2	35	6	Diagonal	1.94E+04	0.046848
2	28	7	Longeron	2.95E+04	0.030534
2	27	8	Diagonal	1.94E+04	0.046848
28	35	9	Longeron	2.95E+04	0.030534
28	27	10	Longeron	2.95E+04	0.030534
27	8	11	Diagonal	1.94E+04	0.046848
27	34	12	Longeron	2.95E+04	0.030534
27	35	13	Diagonal	1.94E+04	0.046848
3	4	14	Longeron	2.95E+04	0.030534
3	15	15	Longeron	2.95E+04	0.030534
3	29	16	Longeron	2.95E+04	0.030534
4	15	17	Diagonal	1.94E+04	0.046848
4	16	18	Longeron	2.95E+04	0.030534
4	17	19	Diagonal	1.94E+04	0.046848
4	42	20	Diagonal	1.94E+04	0.046848
4	29	21	Diagonal	1.94E+04	0.046848
4	30	22	Longeron	2.95E+04	0.030534
4	31	23	Diagonal	1.94E+04	0.046848

4	5	24	Longeron	2.95E+04	0.030534
5	31	25	Longeron	2.95E+04	0.030534
5	6	26	Longeron	2.95E+04	0.030534
5	17	27	Longeron	2.95E+04	0.030534
6	31	28	Diagonal	1.94E+04	0.046848
6	32	29	Longeron	2.95E+04	0.030534
6	33	30	Diagonal	1.94E+04	0.046848
6	7	31	Longeron	2.95E+04	0.030534
6	17	32	Diagonal	1.94E+04	0.046848
6	18	33	Longeron	2.95E+04	0.030534
6	19	34	Diagonal	1.94E+04	0.046848
6	44	35	Diagonal	1.94E+04	0.046848
7	33	36	Longeron	2.95E+04	0.030534
7	8	37	Longeron	2.95E+04	0.030534
7	19	38	Longeron	2.95E+04	0.030534
8	33	39	Diagonal	1.94E+04	0.046848
8	34	40	Longeron	2.95E+04	0.030534
8	35	41	Diagonal	1.94E+04	0.046848
8	9	42	Longeron	2.95E+04	0.030534
8	19	43	Diagonal	1.94E+04	0.046848
8	20	44	Longeron	2.95E+04	0.030534
8	21	45	Diagonal	1.94E+04	0.046848
8	46	46	Diagonal	1.94E+04	0.046848
9	35	47	Longeron	2.95E+04	0.030534
9	10	48	Longeron	2.95E+04	0.030534
9	21	49	Longeron	2.95E+04	0.030534
10	35	50	Diagonal	1.94E+04	0.046848
10	36	51	Longeron	2.95E+04	0.030534
10	37	52	Diagonal	1.94E+04	0.046848
10	11	53	Longeron	2.95E+04	0.030534
10	21	54	Diagonal	1.94E+04	0.046848
10	22	55	Longeron	2.95E+04	0.030534
10	23	56	Diagonal	1.94E+04	0.046848
10	48	57	Diagonal	1.94E+04	0.046848
11	37	58	Longeron	2.95E+04	0.030534
11	12	59	Longeron	2.95E+04	0.030534
11	23	60	Longeron	2.95E+04	0.030534
12	37	61	Diagonal	1.94E+04	0.046848
12	38	62	Longeron	2.95E+04	0.030534
12	39	63	Diagonal	1.94E+04	0.046848
12	13	64	Longeron	2.95E+04	0.030534
12	23	65	Diagonal	1.94E+04	0.046848
12	24	66	Longeron	2.95E+04	0.030534
12	25	67	Diagonal	1.94E+04	0.046848
12	50	68	Diagonal	1.94E+04	0.046848
13	39	69	Longeron	2.95E+04	0.030534
13	14	70	Longeron	2.95E+04	0.030534
13	25	71	Longeron	2.95E+04	0.030534
14	40	72	Longeron	2.95E+04	0.030534

14	25	73	Diagonal	1.94E+04	0.046848
14	26	74	Longeron	2.95E+04	0.030534
14	52	75	Diagonal	1.94E+04	0.046848
14	39	76	Diagonal	1.94E+04	0.046848
29	30	77	Longeron	2.95E+04	0.030534
29	15	78	Diagonal	1.94E+04	0.046848
29	41	79	Longeron	2.95E+04	0.030534
29	42	80	Diagonal	1.94E+04	0.046848
30	42	81	Longeron	2.95E+04	0.030534
30	31	82	Longeron	2.95E+04	0.030534
31	42	83	Diagonal	1.94E+04	0.046848
31	43	84	Longeron	2.95E+04	0.030534
31	17	85	Diagonal	1.94E+04	0.046848
31	44	86	Diagonal	1.94E+04	0.046848
31	32	87	Longeron	2.95E+04	0.030534
32	44	88	Longeron	2.95E+04	0.030534
32	33	89	Longeron	2.95E+04	0.030534
33	44	90	Diagonal	1.94E+04	0.046848
33	45	91	Longeron	2.95E+04	0.030534
33	19	92	Diagonal	1.94E+04	0.046848
33	46	93	Diagonal	1.94E+04	0.046848
33	34	94	Longeron	2.95E+04	0.030534
34	46	95	Longeron	2.95E+04	0.030534
34	35	96	Longeron	2.95E+04	0.030534
35	46	97	Diagonal	1.94E+04	0.046848
35	47	98	Longeron	2.95E+04	0.030534
35	21	99	Diagonal	1.94E+04	0.046848
35	48	100	Diagonal	1.94E+04	0.046848
35	36	101	Longeron	2.95E+04	0.030534
36	48	102	Longeron	2.95E+04	0.030534
36	37	103	Longeron	2.95E+04	0.030534
37	48	104	Diagonal	1.94E+04	0.046848
37	49	105	Longeron	2.95E+04	0.030534
37	23	106	Diagonal	1.94E+04	0.046848
37	50	107	Diagonal	1.94E+04	0.046848
37	38	108	Longeron	2.95E+04	0.030534
38	50	109	Longeron	2.95E+04	0.030534
38	39	110	Longeron	2.95E+04	0.030534
39	50	111	Diagonal	1.94E+04	0.046848
39	51	112	Longeron	2.95E+04	0.030534
39	25	113	Diagonal	1.94E+04	0.046848
39	52	114	Diagonal	1.94E+04	0.046848
39	40	115	Longeron	2.95E+04	0.030534
40	52	116	Longeron	2.95E+04	0.030534
15	41	117	Longeron	2.95E+04	0.030534
15	42	118	Diagonal	1.94E+04	0.046848
15	16	119	Longeron	2.95E+04	0.030534
16	42	120	Longeron	2.95E+04	0.030534
16	17	121	Longeron	2.95E+04	0.030534

17	42	122	Diagonal	1.94E+04	0.046848
17	43	123	Longeron	2.95E+04	0.030534
17	44	124	Diagonal	1.94E+04	0.046848
17	18	125	Longeron	2.95E+04	0.030534
18	44	126	Longeron	2.95E+04	0.030534
18	19	127	Longeron	2.95E+04	0.030534
19	44	128	Diagonal	1.94E+04	0.046848
19	45	129	Longeron	2.95E+04	0.030534
19	46	130	Diagonal	1.94E+04	0.046848
19	20	131	Longeron	2.95E+04	0.030534
20	46	132	Longeron	2.95E+04	0.030534
20	21	133	Longeron	2.95E+04	0.030534
21	46	134	Diagonal	1.94E+04	0.046848
21	47	135	Longeron	2.95E+04	0.030534
21	48	136	Diagonal	1.94E+04	0.046848
21	22	137	Longeron	2.95E+04	0.030534
22	48	138	Longeron	2.95E+04	0.030534
22	23	139	Longeron	2.95E+04	0.030534
23	48	140	Diagonal	1.94E+04	0.046848
23	49	141	Longeron	2.95E+04	0.030534
23	50	142	Diagonal	1.94E+04	0.046848
23	24	143	Longeron	2.95E+04	0.030534
24	50	144	Longeron	2.95E+04	0.030534
24	25	145	Longeron	2.95E+04	0.030534
25	50	146	Diagonal	1.94E+04	0.046848
25	51	147	Longeron	2.95E+04	0.030534
25	52	148	Diagonal	1.94E+04	0.046848
25	26	149	Longeron	2.95E+04	0.030534
26	52	150	Longeron	2.95E+04	0.030534
41	42	151	Longeron	2.95E+04	0.030534
42	43	152	Longeron	2.95E+04	0.030534
43	44	153	Longeron	2.95E+04	0.030534
44	45	154	Longeron	2.95E+04	0.030534
45	46	155	Longeron	2.95E+04	0.030534
46	47	156	Longeron	2.95E+04	0.030534
47	48	157	Longeron	2.95E+04	0.030534
48	49	158	Longeron	2.95E+04	0.030534
49	50	159	Longeron	2.95E+04	0.030534
50	51	160	Longeron	2.95E+04	0.030534
51	52	161	Longeron	2.95E+04	0.030534

APPENDIX D. LABORATORY EXPERIMENTAL TEST LOG

This appendix includes a laboratory log kept of the different modal testing scenarios.

```
Log entry dtd 8-13-97
all files stored in dir:  truss1

all data taken on 8-13, stored as filenames:
test1.mat
test2.mat
test3.mat
test4.mat
test5.mat

Accelerometer placement as follows:
Accel S/N          Node #       Cable #
C112865            3            5
C112400            4            7
C112867            15           6
C112399            16           8
C112866            29           1
C112401            30           3
C112868            41           2
C112398            42           4

axis alignment:
accel truss
+x    +x
+y    -z
+z    +y

impact node used is node #41
(impact equally along all three axis in direction +x, -y, +z (truss
coord.))

Notes:  For testing on 8-13 (filenames test1 to test5), nodal dummy
masses were not used.   Additionally, impact nodes (#41 & #24) had an
extra thumb screw, as an impact target (figure extra mass).  Mounting
table (Newport) was not damped for this test.

*********************************************************
Log entry dtd 8-15-97
all files stored in dir: truss1

all data taken on 8-15, stored as filenames:
(-a.mat is for impact node #4, -b.mat is for impact node #24)
test6a.mat
test7a.mat
test8a.mat
test9a.mat
test10a.mat
test11b.mat
```

test11b.mat
test12b.mat
test13b.mat
test14b.mat
test15b.mat

Accel. placement as follows:

Accel S/N	Node #	Cable #
C112865	3	5
C112400	4	7
C112867	15	6
C112399	16	8
C112866	29	1
C112401	30	3
C112868	41	2
C112398	42	4

Notes: For testing on 8-15, all nodes w/o accels. had dummy mass (approx. 11.5 g) attached. Nodes #41 & #24 each had one extra thumbscrew attached (12.8 g). Newport table not damped for this test.

****

Log entry dtd 8-21-97 AM
all files stored in dir: truss1

all data taken on 8-21 AM, stored as filenames:
(-a.mat is for impact node #4, -b.mat is for impact node #24)
test16a.mat
test17a.mat
test18a.mat
test19a.mat
test20a.mat
test21b.mat
test22b.mat
test23b.mat
test24b.mat
test25b.mat

Accel. placement as follows:

Accel S/N	Node #	Cable #
C112865	3	5
C112400	4	7
C112867	15	6
C112399	16	8
C112866	29	1
C112401	30	3
C112868	41	2
C112398	42	4

Notes: For testing on 8-21 AM, all nodes w/o accels. had dummy mass (approx. 11.5 g) attached. Nodes #41 & #24 each had one extra thumbscrew attached (12.8 g). Newport table WAS damped for this test. dSpace settings were for 10kHz sampling over 0.5 sec. Additionally, several accels were overloaded (plug #7, 21, 22, 19, 9, 10 ,11) slightly when using node #41.

****

Log entry dtd 8-21-97 PM
all files stored in dir: truss1

all data taken on 8-21 PM, stored as filenames:
(-a.mat is for impact node #4, -b.mat is for impact node #24)
test26a.mat
test27a.mat
test28a.mat
test29a.mat
test30a.mat
test31b.mat
test32b.mat
test33b.mat
test34b.mat
test35b.mat

Accel. placement as follows:

Accel S/N	Node #	Cable #
C112865	5	5
C112400	6	7
C112867	17	6
C112399	18	8
C112866	31	1
C112401	32	3
C112868	43	2
C112398	44	4

Notes: For testing on 8-21 PM, all nodes w/o accels. had dummy mass (approx. 11.5 g) attached. Nodes #41 & #24 each had one extra thumbscrew attached (12.8 g). Newport table WAS damped for this test. dSpace settings were for 10kHz sampling over 0.5 sec.

****
Log entry dtd 8-21-97 PM2
all files stored in dir: truss1

all data taken on 8-21 PM2, stored as filenames:
(-a.mat is for impact node #4, -b.mat is for impact node #24)
test36a.mat
test37a.mat
test38a.mat
test39a.mat
test40a.mat
test41b.mat
test42b.mat
test43b.mat
test44b.mat
test45b.mat

Accel. placement as follows:

Accel S/N	Node #	Cable #
C112865	7	5
C112400	8	7
C112867	19	6
C112399	20	8
C112866	33	1
C112401	34	3

C112868	45	2
C112398	46	4

Notes: For testing on 8-21 PM2, all nodes w/o accels. had dummy mass (approx. 11.5 g) attached. Nodes #41 & #24 each had one extra thumbscrew attached (12.8 g). Newport table WAS damped for this test. dSpace settings were for 10kHz sampling over 0.5 sec. (Check out file odd.mat, channel 22)

****

Log entry dtd 8-22-97 (11:45)
all file stored in dir: truss1

all data taken on 8-22 (11:45), stored as filenames:
(-a.mat is for impact node #4, -b.mat is for impact node #24)
test46a.mat
test47a.mat
test48a.mat
test49a.mat
test50a.mat
test51b.mat
test52b.mat
test53b.mat
test54b.mat
test55b.mat

Attention: Accel. order is now switched.
The C50s are on the +x side (truss coord.) of the C10s

Accel. placement as follows:

Accel S/N	Node #	Cable #
C112865	10	5
C112400	9	7
C112867	22	6
C112399	21	8
C112866	36	1
C112401	35	3
C112868	48	2
C112398	47	4

Notes: For testing on 8-22 (11:45), all nodes w/o accels. had dummy mass (approx. 11.5 g) attached. Nodes #41 & #24 each had one extra thumbscrew attached (12.8 g). Newport table WAS damped for this test. dSpace settings were for 10kHz sampling over 0.5 sec.

****

Log entry dtd 8-22-97 (12:35)
all file stored in dir: truss1

all data taken on 8-22 (12:35), stored as filenames:
(-a.mat is for impact node #4, -b.mat is for impact node #24)
test56a.mat
test57a.mat
test58a.mat
test59a.mat
test60a.mat
test61b.mat

test62b.mat
test63b.mat
test64b.mat
test65b.mat

Attention: Accel. order is now switched.
The C50s are on the +x side (truss coord.) of the C10s

Accel. placement as follows:

Accel S/N	Node #	Cable #
C112865	12	5
C112400	11	7
C112867	24	6
C112399	23	8
C112866	38	1
C112401	37	3
C112868	50	2
C112398	49	4

Notes: For testing on 8-22 (12:35), all nodes w/o accels. had dummy mass (approx. 11.5 g) attached. Nodes #41 & #24 each had one extra thumbscrew attached (12.8 g). Newport table WAS damped for this test. dSpace settings were for 10kHz sampling over 0.5 sec.
test63b, bias channel 23 consistent overloading on x axis on the 10g accels.

****
Log entry dtd 8-22-97 (13:25)
all file stored in dir: truss1

all data taken on 8-22 (13:25), stored as filenames:
(-a.mat is for impact node #4, -b.mat is for impact node #24)
test66a.mat
test67a.mat
test68a.mat
test69a.mat
test70a.mat
test71b.mat
test72b.mat
test73b.mat
test74b.mat
test75b.mat

Attention: Accel. order is now switched.
The C50s are on the +x side (truss coord.) of the C10s

Accel. placement as follows:

Accel S/N	Node #	Cable #
C112865	14	5
C112400	13	7
C112867	26	6
C112399	25	8
C112866	40	1
C112401	39	3
C112868	52	2
C112398	51	4

Notes: For testing on 8-22 (13:25), all nodes w/o accels. had dummy mass (approx. 11.5 g) attached. Nodes #41 & #24 each had one extra thumbscrew attached (12.8 g). Newport table WAS damped for this test. dSpace settings were for 10kHz sampling over 0.5 sec.
test -a.mat no bias for chan. 23
test -b.mat tiny bias for chan. 23

****
Log entry dtd 8-22-97 (14:55)
all file stored in dir: truss1

all data taken on 8-22 (14:55), stored as filenames:
(-a.mat is for impact node #4, -b.mat is for impact node #24)
test76a.mat
test77a.mat
test78a.mat
test79a.mat
test80a.mat

Attention: All accels. now in a line (along x axis, upper y, neg z, truss coord.) Accels. start at node52-45

Accel. placement as follows:

Accel S/N	Node #	Cable #
C112865	50	5
C112400	47	7
C112867	45	6
C112399	46	8
C112866	52	1
C112401	49	3
C112868	51	2
C112398	48	4

Notes: For testing on 8-22 (14:55), all nodes w/o accels. had dummy mass (approx. 11.5 g) attached. Nodes #41 & #24 each had one extra thumbscrew attached (12.8 g). Newport table WAS damped for this test. dSpace settings were for 10kHz sampling over 0.5 sec.
test -a.mat no bias for chan. 23
test -b.mat tiny bias for chan. 23

****
Log entry dtd 8-22-97 (16:25)
all file stored in dir: truss1

all data taken on 8-22 (16:25), stored as filenames:
(-a.mat is for impact node #4, -b.mat is for impact node #24)
test81a.mat
test82a.mat
test83a.mat
test84a.mat
test85a.mat
test86b.mat
test87b.mat
test88b.mat
test89b.mat
test90b.mat

Attention: Global test (maximum spread of accels.)

Accel. placement as follows:

Accel S/N	Node #	Cable #
C112865	3	5
C112400	44	7
C112867	14	6
C112399	20	8
C112866	52	1
C112401	49	3
C112868	41	2
C112398	11	4

Notes: For testing on 8-22 (16:25), all nodes w/o accels. had dummy mass (approx. 11.5 g) attached. Nodes #41 & #24 each had one extra thumbscrew attached (12.8 g). Newport table WAS damped for this test. dSpace settings were for 10kHz sampling over 0.5 sec.
test -b.mat slightly satu. chan. 7

APPENDIX E. ELECTRONIC HARDWARE DOCUMENTATION

The following values pertain to the Kistler Instrument Corp. accelerometers and signal conditioners, and the PCB® Piezotronics impulse force hammer and signal conditioner that were used in the modal testing and analysis of the NPS Space Truss.

Kistler Instrument Corp. Accelerometers:
(Note: $g = 9.807\ m/s^2$)

		Sensitivity at 100 Hz, 3 g rms			
Type	Serial Number	+ x-axis	+ y-axis	+ z-axis	
8690C50	C112865	98.7	101.6	97.7	mV/g
8690C50	C112866	101.1	100.3	96.7	mV/g
8690C50	C112867	98.9	99.8	99.7	mV/g
8960C50	C112868	99.2	99.5	99.1	mV/g
8690C10	C112398	495	490	494	mV/g
8690C10	C112399	487	490	490	mV/g
8690C10	C112400	499	500	494	mV/g
8690C10	C112401	497	491	505	mV/g

Kistler Instrument Corp. Signal Conditioners (Multi-Channel Couplers):

Type	Serial Number
5124A (twelve channel)	C74930
5124A (twelve channel)	C74929

PCB® Piezotronics Impulse Force Hammer:

Type	Serial Number	Notes
086B01	4144	Hammer config.: hard plastic tip w/ tuning mass

PCB® Piezotronics Signal Conditioner:

Type	Serial Number	Notes
484B	2086	Set CPLG to AC & Bias to 11 V

Micron Optics, Fiber Bragg Grating Interrogation System (FBG-IS):

Type	Serial Number	Notes
picoWave	3005	Version 3.0

APPENDIX F. IMPORTANT POINTS OF CONTACT

The follow points of contact are worth listing as they are able to provide valuable support in regards to the purchase, operation, and maintenance of the NPS Space Truss, test equipment, and fiber optic equipment.

Name	Association	Phone Number
Albert Bosse, Ph.D.	Naval Research Lab	(202) 404 2724 (lab)
		(202) 404 8341 (office)
		(202) 767 9339 (FAX)
Tom Li	Micron Optics, Inc.	(408) 374 8664 (office)
		(408) 374 1461 (FAX)
Nick Plescia	Bragg Photonics	(514) 421 6766 (office)
		(514) 421 0560 (FAX)
James Borkowski	Kistler Instrument Corp.	(716) 691 5100 (office)

APPENDIX G. XFER.M – MATLAB ANALYSIS CODE

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% xfer.m
% Allows multiple tfe.m (transfer function estimates) on one plot
% NOTE: LOAD TEST DATA PRIOR TO RUNNING (use load function)
%       see APPENDIX D, LABORATORY EXPERIMENTAL TEST LOG for filenames
% User provides channels to be analyzed (up to 8 per plot)
% xfer.m scales the input(impulse hammer) and output(accelerometers)
% last revision:  970910, Brent K. Andberg for the NPS Space Truss
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Gathers data from user %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
i = input('Enter number of channels to analyze (up to 8): ');

for p = 1:i;
      chn(p) = input(['Enter channel ', int2str(p), ': ']);
end

n = input('Enter figure number: ');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% accelerometer sensitivity values [mV/g] %%%%%%%%%%%%%%%%%%%%%%
% digits correspond to last two #s in accelerometer s/n
% letter (x, y, or z) corresponds to accelerometer's axis
a65x = 98.8;  a65y = 101.6; a65z = 97.7;
a66x = 101.1; a66y = 100.3; a66z = 96.7;
a67x = 98.9;  a67y = 99.8;  a67z = 99.7;
a68x = 99.2;  a68y = 99.5;  a68z = 99.1;
a98x = 495;   a98y = 490;   a98z = 494;
a99x = 487;   a99y = 490;   a99z = 490;
a00x = 499;   a00y = 500;   a00z = 494;
a01x = 497;   a01y = 491;   a01z = 505;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% scale factor for hammer impact alignment
for q = 1:i;
  if chn(q) == 3
      amp(q) = 0.707;
      elseif chn(q) == 6
      amp(q) = 0.707;
      elseif chn(q) == 9
      amp(q) = 0.707;
      elseif chn(q) == 12
      amp(q) = 0.707;;
      elseif chn(q) == 15
      amp(q) = 0.707;
      elseif chn(q) == 18
      amp(q) = 0.707;
      elseif chn(q) == 21
      amp(q) = 0.707;
      elseif chn(q) == 24
      amp(q) = 0.707;
      else
```

```
      amp(q) = 0.5;
  end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% scale factor for accelerometer sensitivity
for r = 1:i;
  if chn(r) == 1
      accel(r) = a66x;
      elseif chn(r) == 2
      accel(r) = a66y;
      elseif chn(r) == 3
      accel(r) = a66z;
      elseif chn(r) == 4
      accel(r) = a68x;
      elseif chn(r) == 5
      accel(r) = a68y;
      elseif chn(r) == 6
      accel(r) = a68z;
      elseif chn(r) == 7
      accel(r) = a01x;
      elseif chn(r) == 8
      accel(r) = a01y;
      elseif chn(r) == 9
      accel(r) = a01z;
      elseif chn(r) == 10
      accel(r) = a98x;
      elseif chn(r) == 11
      accel(r) = a98y;
      elseif chn(r) == 12
      accel(r) = a98z;
      elseif chn(r) == 13
      accel(r) = a65x;
      elseif chn(r) == 14
      accel(r) = a65y;
      elseif chn(r) == 15
      accel(r) = a65z;
      elseif chn(r) == 16
      accel(r) = a67x;
      elseif chn(r) == 17
      accel(r) = a67y;
      elseif chn(r) == 18
      accel(r) = a67z;
      elseif chn(r) == 19
      accel(r) = a00x;
      elseif chn(r) == 20
      accel(r) = a00y;
      elseif chn(r) == 21
      accel(r) = a00z;
      elseif chn(r) == 22
      accel(r) = a99x;
      elseif chn(r) == 23
      accel(r) = a99y;
      else
      accel(r) = a99z;
  end
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% stores tfe data into a matrix for multiple plots (after scaling)
% tfe is set for <5001 data points, and a samp. freq of 10k
figure(n);
clf;
for s = 1:i;
[x(:,s) y(:,s)] =
tfe(amp(s)*trace_y(25,:),(1/(0.001*accel(s)))*trace_y(chn(s),:),4096,100
00);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% displays data, complete w/ grid, legend, and title, etc.
hold on;
for t = 1:i;
      plot(y(:,t), 20*log10(x(:,t)), sprintf('c%1.0f',t))
end

axis([0 300 -20 100])
grid

if i == 1
 legend(int2str(chn(1)));
 elseif i == 2
 legend(int2str(chn(1)),int2str(chn(2)));
 elseif i == 3
 legend(int2str(chn(1)),int2str(chn(2)),int2str(chn(3)));
 elseif i == 4

legend(int2str(chn(1)),int2str(chn(2)),int2str(chn(3)),int2str(chn(4)));
 elseif i == 5

legend(int2str(chn(1)),int2str(chn(2)),int2str(chn(3)),int2str(chn(4)),i
nt2str(chn(5)));
 elseif i == 6

legend(int2str(chn(1)),int2str(chn(2)),int2str(chn(3)),int2str(chn(4)),i
nt2str(chn(5)),int2str(chn(6)));
 elseif i == 7

legend(int2str(chn(1)),int2str(chn(2)),int2str(chn(3)),int2str(chn(4)),i
nt2str(chn(5)),int2str(chn(6)),int2str(chn(7)));
 else

legend(int2str(chn(1)),int2str(chn(2)),int2str(chn(3)),int2str(chn(4)),i
nt2str(chn(5)),int2str(chn(6)),int2str(chn(7)),int2str(chn(8)));
end

title('Transfer Funct. Estimate (see legend for channel #s)');
ylabel('Transfer Function Estimate (dB)');
xlabel('Frequency (Hz)');
hold off;
```

APPENDIX H. OVERLAY.M – MATLAB ANALYSIS CODE

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% overlay.m
% After running the m-file: xfer.m, running overlay.m will
% superimpose an overlay of the NPS Space Truss's
% natural freqs. (0-300 Hz)
% over the selected experimental data (user chooses figure window)
% By Brent K. Andberg, for the NPS Space Truss
% last revision, 970914
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Gathers data from user %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

n = input('Enter figure number: ');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% displays computed nat. freqs. [Hz]
hold on;

freq = [14.64 16.26 30.41 33.97 62.93 74.54 80.66 101.01 126.23 135.97
140.96 198.44 207.73 229.59 232.66 257.27 280.48 284.65];

for ii = 1:18;
  for jj = 1:121;
    freq1(jj) = freq(ii);
  end
  plot(freq1, (-20:100), 'g-');
end

hold off;
```

APPENDIX I. DATA PLOTS

This appendix contains an extensive collection of truss data displayed via MATLAB plots using the *xfer.m* code in conjunction with the *overlay.m* code. Filenames with the *–a.mat* suffix use node 41 as an impact point and those with *–b.mat* as a suffix used node 24. For each test selected, the x, y, and z axes are displayed for comparison. Refer to Appendix D, Laboratory Experimental Test Log, for information on accelerometer placement. All data was collected using a 10,000 Hz sampling frequency, over a period of 0.5 second. The vertical, green lines superimposed over the plot are representative of the computed natural frequencies of the NPS Space Truss.

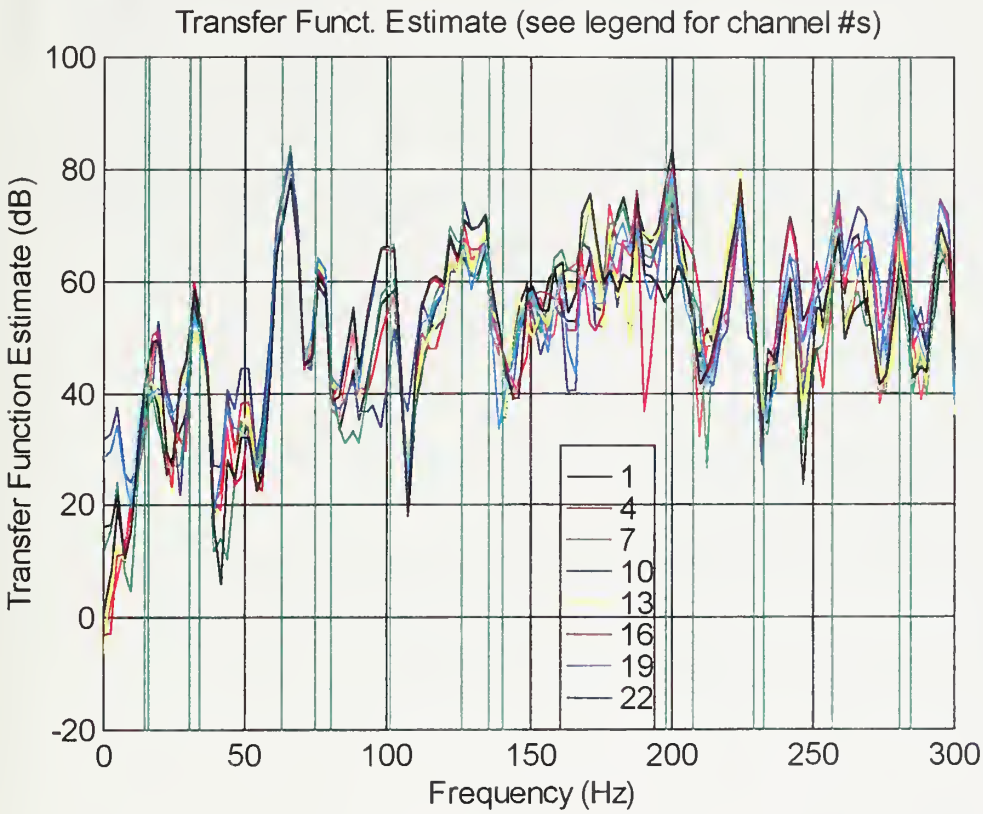


Figure 30. Plot of *test66a.mat* (x-axis)

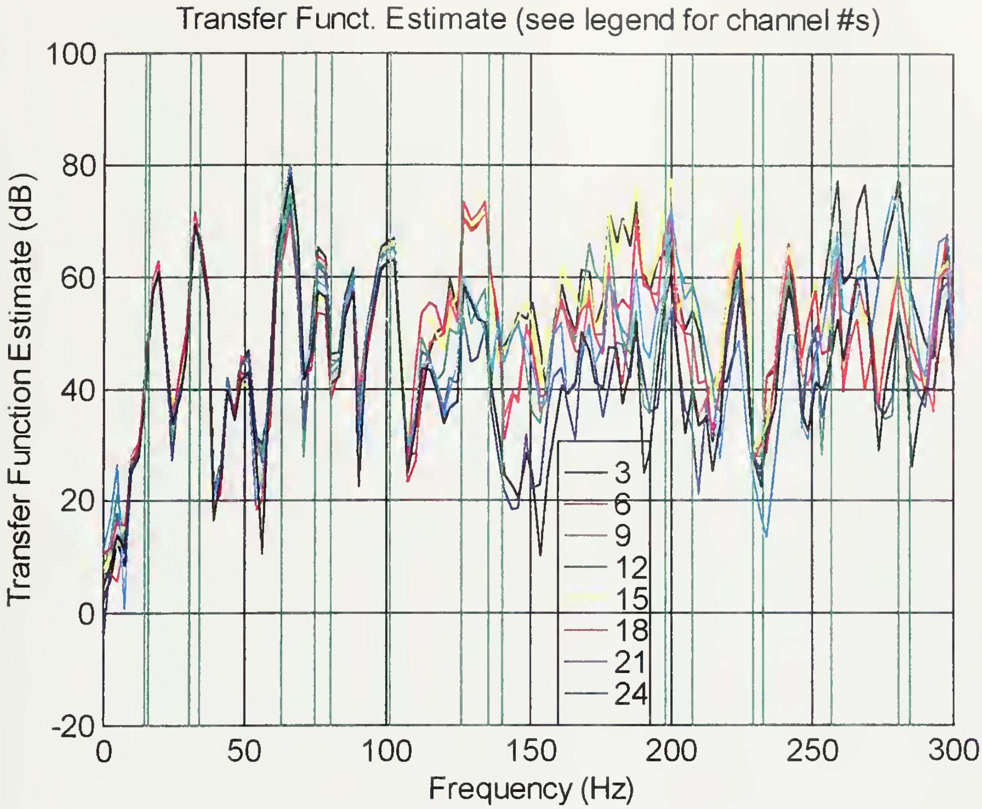


Figure 31. Plot of *test66a.mat* (y-axis)

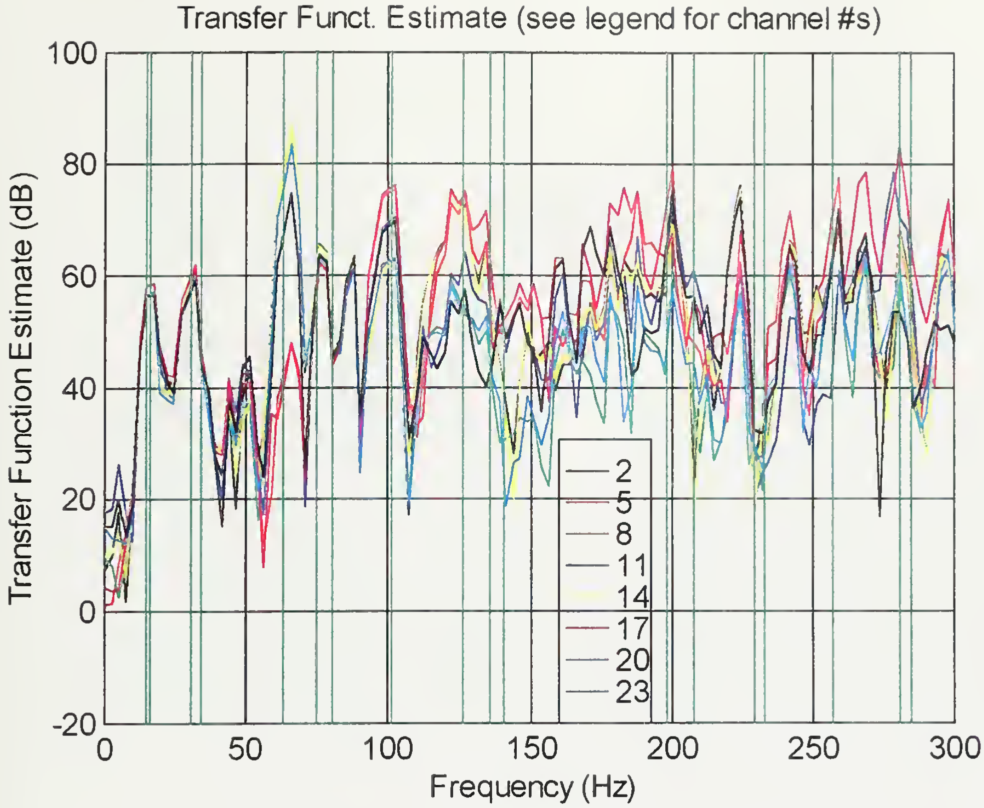


Figure 32. Plot of *test66a.mat* (z-axis)

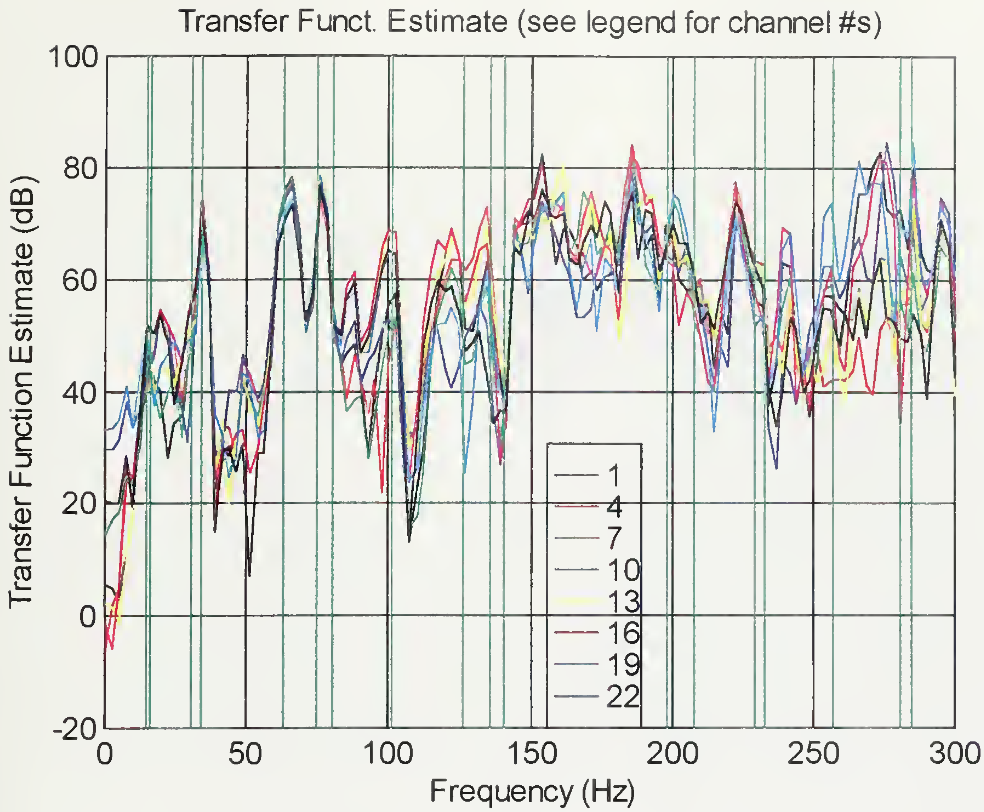


Figure 33. Plot of *test57a.mat* (x-axis)

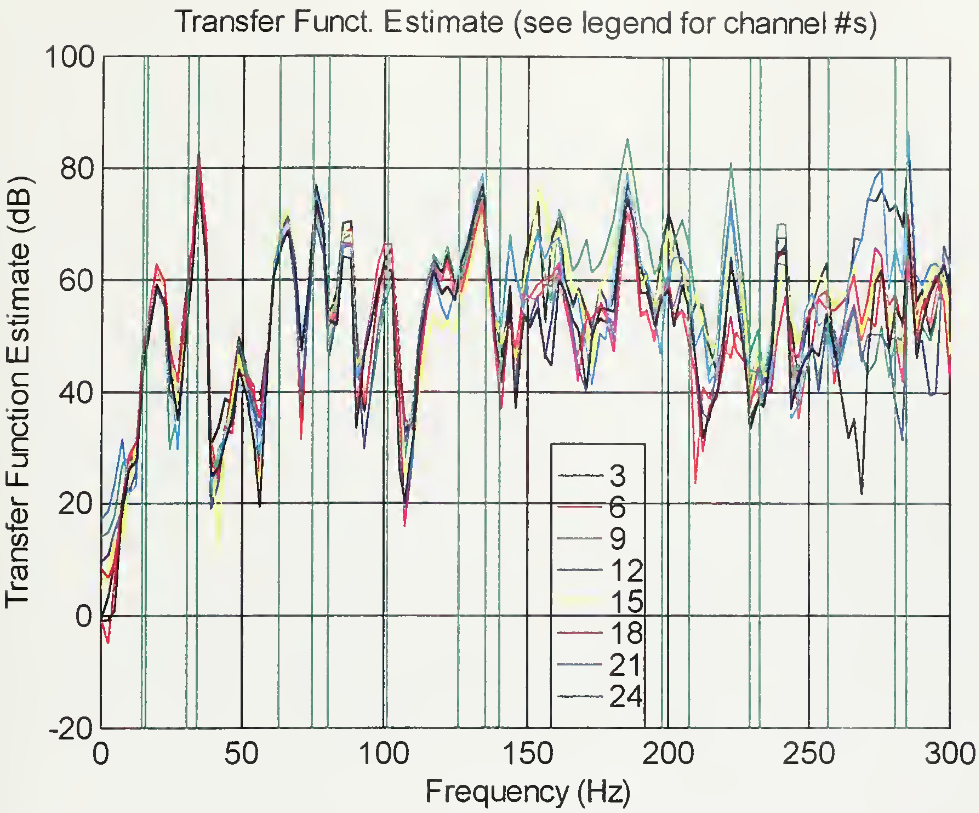


Figure 34. Plot of *test57a.mat* (y-axis)

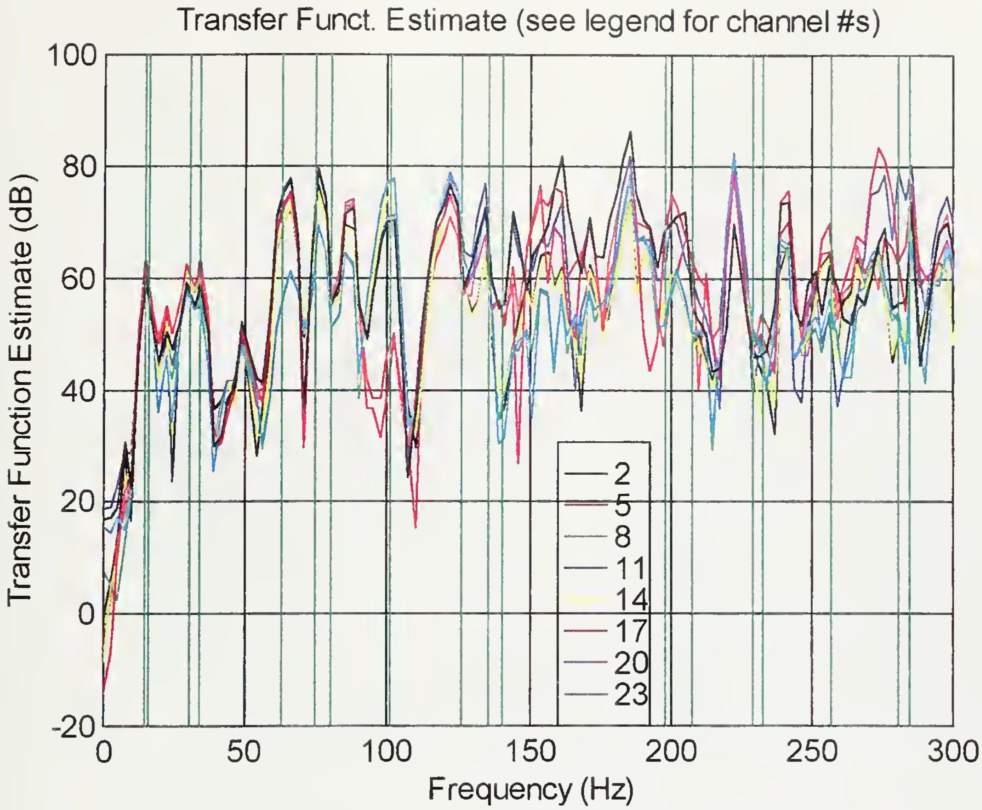


Figure 35. Plot of *test57.mat* (z-axis)

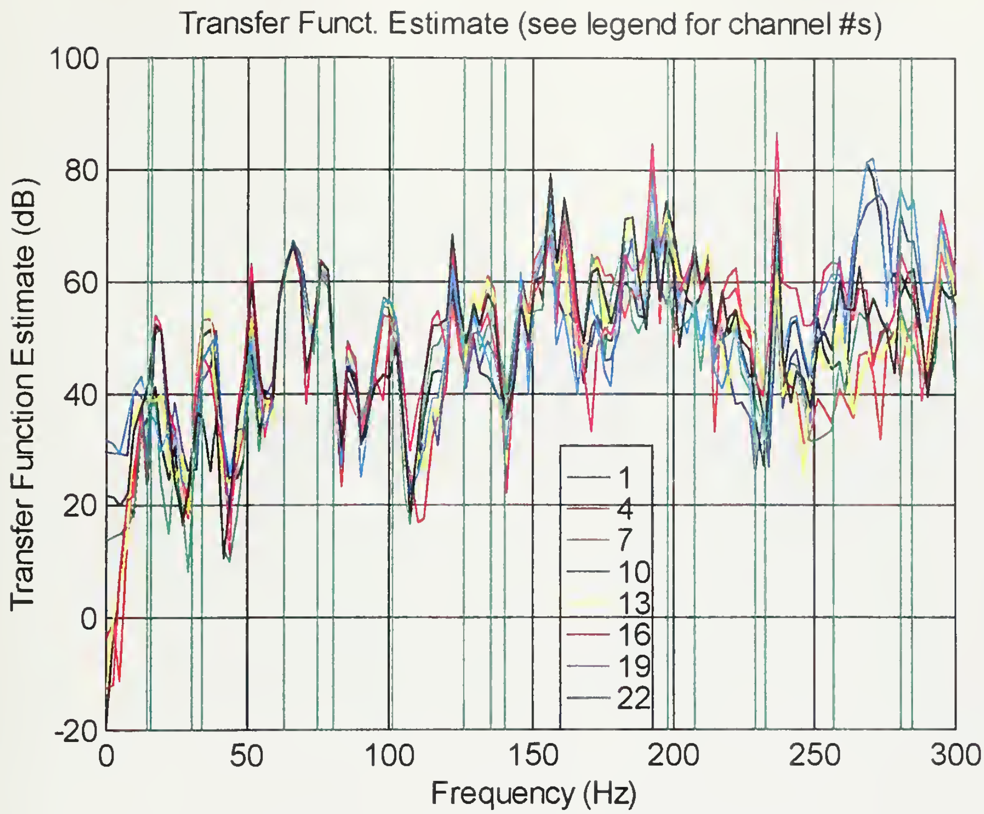


Figure 36. Plot of *test48a.mat* (x-axis)

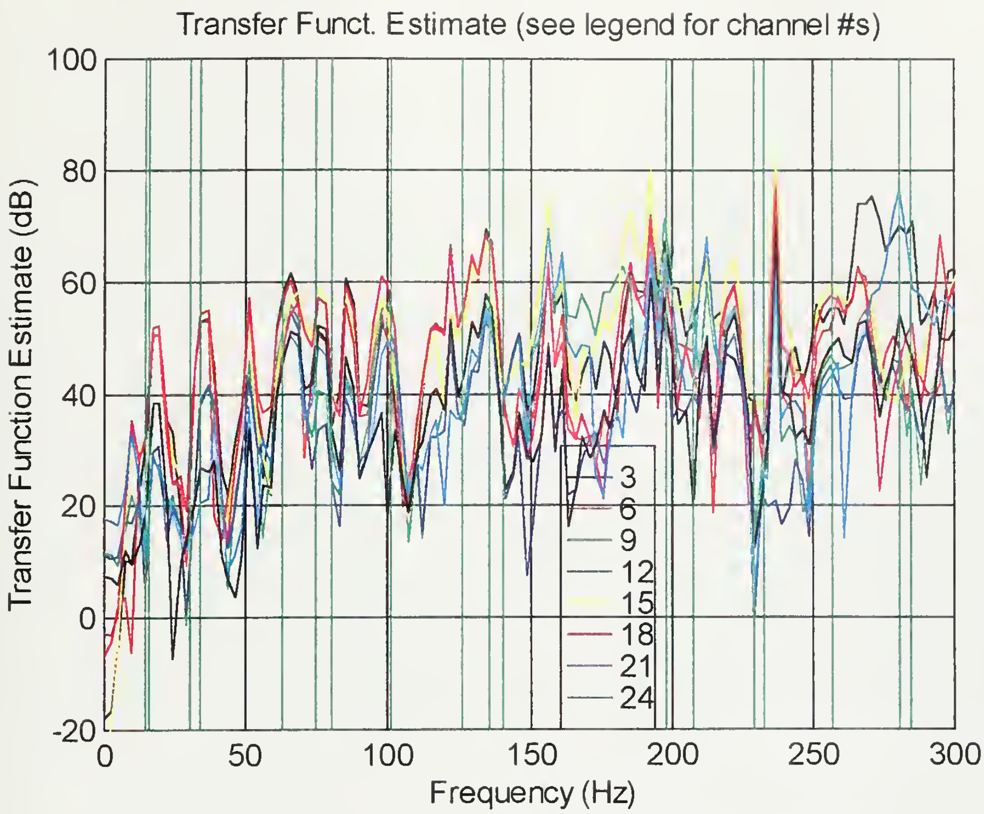


Figure 37. Plot of *test48a.mat* (y-axis)

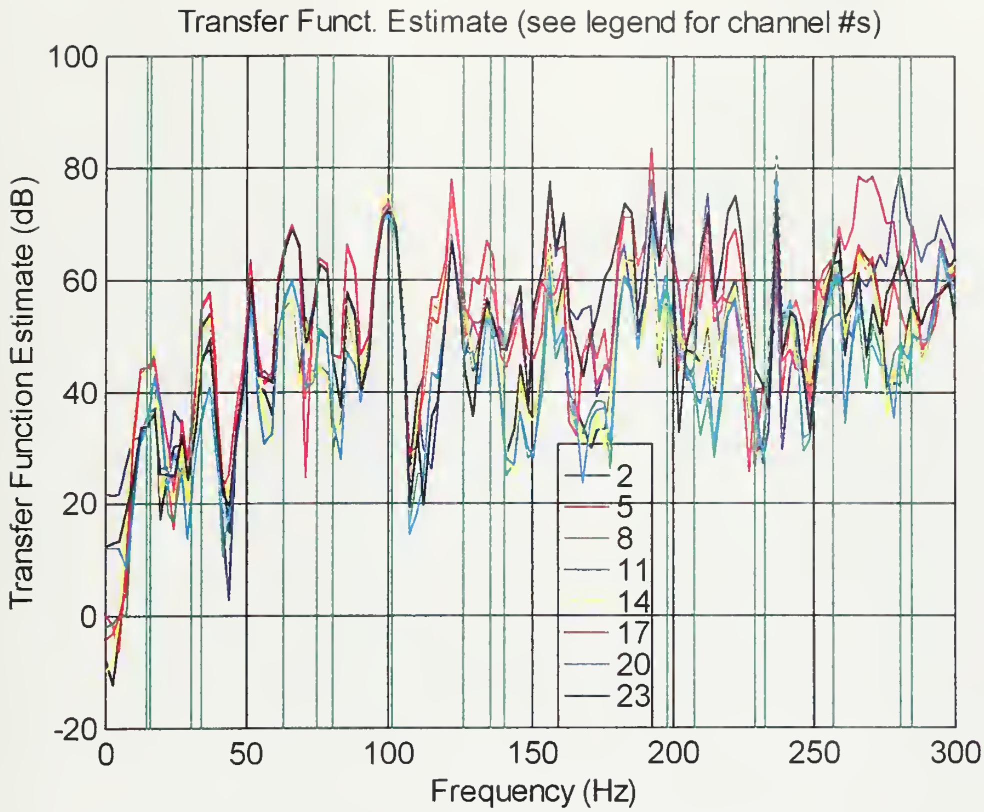


Figure 38. Plot of *test48a.mat* (z-axis)

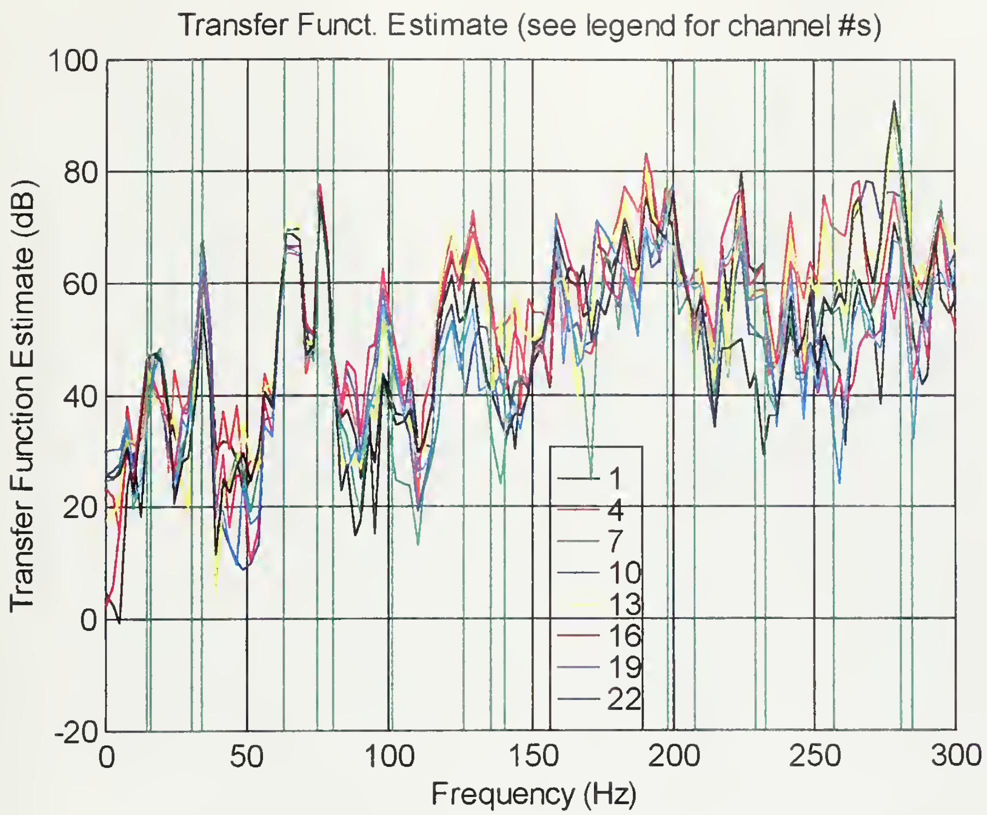


Figure 39. Plot of *test81a.mat* (x-axis)

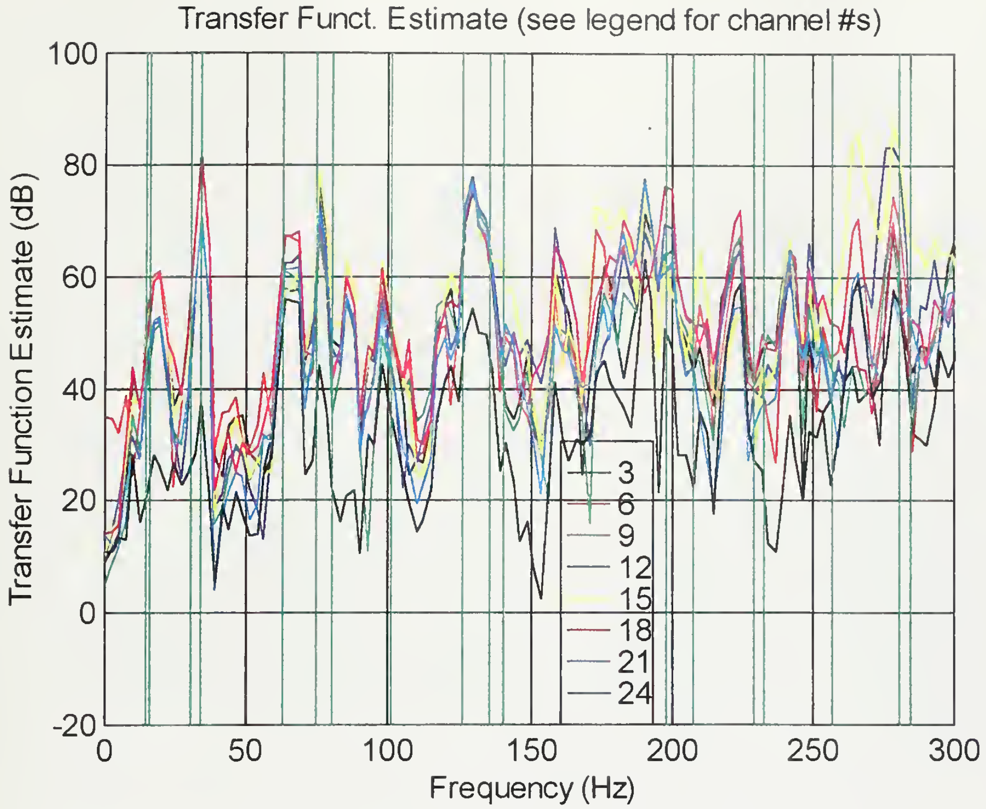


Figure 40. Plot of *test81a.mat* (y-axis)

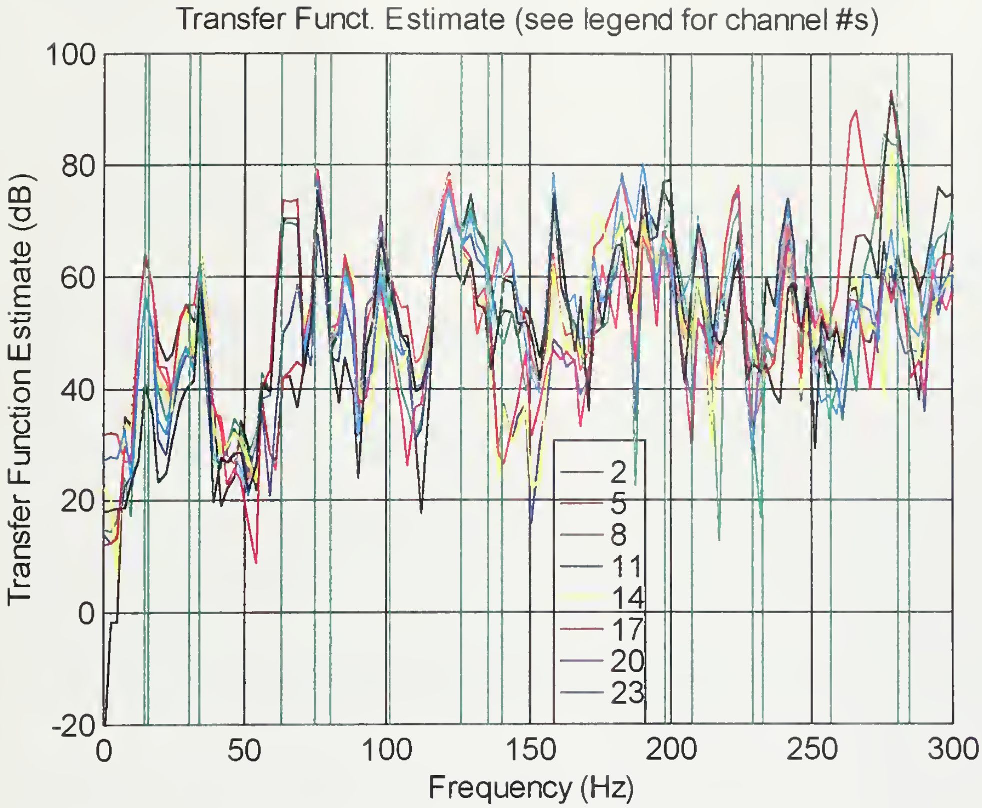


Figure 41. Plot of *test81a.mat* (z-axis)

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center.................................. 2
   8725 John J. Kingman Rd., STE 0944
   Ft. Belvoir, Virginia 22060-6218

2. Dudley Knox Library.. 2
   Naval Postgraduate School
   411 Dyer Rd.
   Monterey, California 93943-5101

3. Director, Training and Education.. 1
   MCCDC, Code C46
   1019 Elliot Rd.
   Quantico, Virginia 22134-5027

4. Director, Marine Corps Research Center................................ 2
   MCCDC, Code C40RC
   2040 Broadway Street
   Quantico, Virginia 22134-5107

5. Director, Studies and Analysis Division.................................. 1
   MCCDC, Code C45
   300 Russell Road
   Quantico, Virginia 22134-5130

6. Marine Corps Representative... 1
   Naval Postgraduate School
   Code 037, Bldg. 234, HA-220
   699 Dyer Road
   Monterey, California 93940

7. Marine Corps Tactical Systems Support Activity........................1
Technical Advisory Branch
Attn: Maj. J.C. Cummiskey
Box 555171
Camp Pendleton, California 92055-5080

8. Chairman, Code AA..1
Department of Aeronautics and Astronautics
Naval Postgraduate School
Monterey, California 93943

9. Professor Brij N. Agrawal, Code AA/Ag................................ 2
Department of Aeronautics and Astronautics
Naval Postgraduate School
Monterey, California 93943

10. Dr. Gangbing Song, Code AA/Sb..2
Department of Aeronautics and Astronautics
Naval Postgraduate School
Monterey, California 93943

11. CAPT Brent K. Andberg, USMC... 3
3831 Windhurst Drive
Lilburn, Georgia 30047

12. Dr. Albert Bosse... 1
Naval Research Laboratory
Code 8220
Washington, D.C. 20375-5355

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY CA 93943-5101

DUDLEY KNOX LIBRARY
3 2768 00340667 9